NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

ON SYSTEMS ENGINEERING PROCESSES IN SYSTEM-OF-SYSTEMS ACQUISITION
by
Heng Jiin Shyang
June 2011
Thesis Advisor: Thomas V. Huynh
Second Reader: John S. Osmundson

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2011	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE On Systems Engineering Processes in System-of Systems Acquisition			**5. FUNDING NUMBERS**
6. AUTHOR(S) Heng Jiin Shyang			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB protocol number: N/A			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

System-of-Systems (SoS) programs applying the current systems engineering (SE) process in their acquisition have met with numerous technical and program management challenges resulting in adverse consequences such as unacceptable schedule delays. To enhance the chance of success for SoS acquisition, the current acquisition process needs to be improved. Systems engineering has been a recognized contributor to successful systems acquisition and its applicability to SoS is apparent. In this research, a proposed SoS SE process comprising extensive front-end SE activities is compared with the current SoS SE process.

The interaction of stakeholders and activities in both the current and proposed SoS SE processes are presented using System Modeling Language (SysML) diagrams. Modeling and Simulation (M&S) is also used to show that the proposed SoS SE process is able to help in reducing the likelihood of schedule delays.

The M&S results show that a low-risk SoS acquisition could continue with the current SE process as the benefits derived from an extensive front-end SE process are limited. Conversely, a high-risk SoS acquisition should adopt the SoS SE process proposed herein to enhance the SoS acquisition program's chance of success. It is high-risk SoS acquisitions such as the US Army's Future Combat System, the US Coast Guard's Deep Water System, the Joint Tactical Radio System (JTRS), and Homeland Security's SBInet that would likely benefit from the proposed SoS SE process.

14. SUBJECT TERMS System of Systems, Acquisition, Systems Engineering, SysML			**15. NUMBER OF PAGES** 91
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

ON SYSTEMS ENGINEERING PROCESSES IN SYSTEM-OF-SYSTEMS ACQUISITION

Heng Jiin Shyang
Civilian, Defence Science & Technology Agency, Singapore
B.Eng., Nanyang Technological University, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2011

Author: Heng Jiin Shyang

Approved by: Professor Thomas V. Huynh, PhD
Thesis Advisor

Professor John S. Osmundson, PhD
Second Reader

Professor Clifford A. Whitcomb, PhD
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

System-of-Systems (SoS) programs applying the current systems engineering (SE) process in their acquisition have met with numerous technical and program management challenges resulting in adverse consequences such as unacceptable schedule delays. To enhance the chance of success for SoS acquisition, the current acquisition process needs to be improved. Systems engineering has been a recognized contributor to successful systems acquisition and its applicability to SoS is apparent. In this research, a proposed SoS SE process comprising extensive front-end SE activities is compared with the current SoS SE process.

The interaction of stakeholders and activities in both the current and proposed SoS SE processes are presented using System Modeling Language (SysML) diagrams. Modeling and Simulation (M&S) is also used to show that the proposed SoS SE process is able to help in reducing the likelihood of schedule delays.

The M&S results show that a low-risk SoS acquisition could continue with the current SE process as the benefits derived from an extensive front-end SE process are limited. Conversely, a high-risk SoS acquisition should adopt the SoS SE process proposed herein to enhance the SoS acquisition program's chance of success. It is high-risk SoS acquisitions such as the US Army's Future Combat System, the US Coast Guard's Deep Water System, the Joint Tactical Radio System (JTRS), and Homeland Security's SBInet that would likely benefit from the proposed SoS SE process.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

A&D	Analysis and Design
DoD	Department of Defense
EMD	Engineering Manufacturing Development
FCS	Future Combat System
GAO	Government Accountability Office
IMP	Integration, Modification and Production
MOE	Measure of Effectiveness
M&S	Modeling and Simulation
Pf	Probability of Failure
Pf(Current)	Probability of Failure of Sys5 and SoS11 in the Current SoS SE Process
Ps	Probability of Success
PO	Program Office
SE	Systems Engineering
SoS	System of Systems
SysML	Systems Modeling Language
TVV	Testing, Verification and Validation

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author would like to express his gratitude towards Dr. Tom Huynh for his constant guidance, encouragement, and patience throughout this research. His vast experience in the field of Systems Engineering has helped the author to gain a deeper understanding of and appreciation for the discipline.

The author is also grateful to Dr. John S. Osmundson for patiently going through the thesis and providing invaluable comments and advice to improve it.

Lastly, the author will also like to thank his wife, Geok Pei, for providing support and encouragement throughout the 18 months of study at the Naval Postgraduate School.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Many definitions of a system of systems (SoS) exist in published literature, but there is currently no universally accepted definition of a SoS (Sage and Biemer, 2007). Huynh and Osmundson (2007) define a SoS as a conglomeration of existing, stand-alone systems and to-be-developed systems that are integrated and interoperable with each other to achieve a capability that the individual systems alone cannot provide. Two systems are interoperable if they can successfully exchange and process information in support of a task or mission. DeLaurentis and Mane (2009) describe a SoS as consisting of multiple, heterogeneous, distributed systems that can (and do) operate independently but can also be assembled in networks and collaborate to achieve a goal. Sage and Biemer (2007) suggest that a SoS is a large-scale, complex system, involving a combination of technologies, humans, and organizations, and consisting of components. The United States (U.S.) Department of Defense (DoD) Systems Engineering (SE) guide for SoS defines a SoS as a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities. (ODUSD(A&T)SSE, 2008, DoD, 2004). It is common that a SoS developed by the U.S. DoD comprises of one or more new systems that need to interoperate with each other and with existing in-service systems to produce a unique capability to satisfy the needs of the user[1].

As the demands for warfare and military operations evolve, there is an increasing need for modern armed forces to acquire complex systems of systems to fulfill their operational needs. This increased demand may have been so rapid that a proper and established acquisition process tailored for systems of systems has yet to be conceived. The lack of such an acquisition process has contributed to the demise of some recent SoS acquisition programs, such as the US Army's Future Combat System, the US Coast Guard's Deep Water System, the Joint Tactical Radio System (JTRS), and Homeland

[1] User refers to the relevant user community, which has vested interest in the SoS.

Security's SBInet (Huynh *et al.*, 2010). These SoS acquisition programs have experienced technical, budget, and schedule challenges beyond what is considered the usual norm for single-system acquisitions.

With an aim to develop approaches that can prevent such SoS acquisition programs from failing, Ghose and DeLaurentis (2008) look into "types of acquisition management, policy insights, and approaches that can increase the success of an acquisition in the SoS setting." They investigate the impact of SoS attributes, such as "requirement interdependency, project risk, and span-of-control of SoS managers and engineers—on the completion time of SoS projects." Using a conceptual model for SoS acquisition activities within the current SE process for acquisition of single systems, DeLaurentis and Mane (2009) find that projects with a high span-of-control always require less time than those with low span-of-control and that the effects of the span-of-control are much more significant as compared with project risks and requirements dependency.

The work in (Huynh *et al.*, 2010) on effective contracting structures and processes for SoS acquisition suggests further that, to maximize the probability of SoS acquisition success, extensive span-of-control by systems engineers be sustained throughout a SoS acquisition—during the pre-acquisition and acquisition phases of the SoS acquisition—and change be made to existing contracting structures and process and organizational structures.

The role of SE has been recognized to be critical in successful systems acquisition (ODUSD(A&T)SSE, 2008). Span-of-control by systems engineers in the pre-acquisition phase amounts to carrying out early SE activities in a SoS acquisition program. Lack of resources and of early and disciplined SE in the acquisition program has been cited as the contributors to poor program outcomes (GAO, 2009). The importance of early SE activities in SoS acquisition is expected to be more pronounced, compared to single-system acquisition, as there are significant differences between a system and a SoS.

Span-of-control is thus an important factor that will enhance the probability of a SoS program meeting its schedule. However, the details on how to implement the span-

of-control in a SoS acquisition are not clear. Currently, there is no known SE process to effect the realization of the span-of-control of engineers and managers during the pre-acquisition and acquisition phases of a SoS acquisition. A new SE process for SoS acquisition is thus needed, and its effectiveness need be quantitatively assessed and compared to that of the SE process currently used in SoS acquisition. The definition of such a SE process and the quantitative assessment of its effectiveness in SoS acquisition motivate this research.

A. RESEARCH QUESTIONS

This research aims to propose and develop a SE process to enhance SoS acquisition success. The following three research questions are investigated:

1. What are the issues affecting the success of SoS acquisition programs? What SE process is currently used for ongoing SoS programs?

2. How will the proposed SE process differ from the current SE process used in SoS acquisition?

3. How will the proposed SE process be quantitatively shown to be better than the current SE process in SoS acquisition?

B. APPROACH

The approach employed to answer the research questions consists of the following three major steps.

1. This research begins by identifying the issues that could have caused the current SoS acquisition process to be prone to failures. Through review of past relevant work, the need to include front-end SE activities and having a span-of-control throughout the SoS acquisition by the SoS program office are shortlisted as possible means to improve the current SE process.

2. Front-end SE activities are identified based on their ability to enhance interactions among stakeholders and produce an over-arching architecture to guide the SoS acquisition. Interfaces among the activities are also identified to

show interactions between activities and stakeholders and the flow of the acquisition process. The new proposed SE process in SoS acquisition, hereinafter called the proposed SoS SE process, is then captured in a process flow model.

3. The proposed SoS SE process is developed, and its effectiveness is compared to that of the current SE process. Systems Modeling Language (SysML) (OMG SysML ™, 2010) provides the means to construct the current and proposed SoS SE models. ExtendSim (a modeling software), a M&S tool, is used to implement the SysML models so that a Monte Carlo simulation can be performed on the two models. The simulation outputs are then used for the comparative analysis.

A discussion of the use of SysML and the modeling and simulation (M&S) approach follows.

1. Systems Modeling Language

SysML is a widely used modeling language for systems engineers. It has the ability to support specification, analysis, design, verification, and validation for an extensive range of complex systems, including hardware, software, information, processes, personnel, and facilities. The aim of SysML is to standardize the different modeling languages used by systems engineers (OMG SysML™, 2010). In this research, three types of behavioral diagrams are used to represent the activities, message sequence, and high-level functionality of the two SE processes (current and proposed).

Based on the SysML activity and sequence diagrams, two SoS acquisition models are developed—the current SoS Acquisition Model and the proposed SoS Acquisition Model. The current SoS Acquisition Model is based on the understanding of the current DoD SoS acquisition process. It starts from the point when the decision authority makes a SoS acquisition decision until the SoS transits to the user for operations and deployment. A set of front-end SE activities are introduced into the current SoS acquisition process to form the proposed SoS acquisition process. The key front-end SE activities for the SoS program office are: deriving final SoS requirements, generating SoS concept alternatives,

and developing, and selecting the final SoS architecture. Constant interactions between the SoS program office and the system program offices is another feature of the proposed SoS acquisition process.

2. Modeling and Simulation

Modeling and Simulation (M&S) is the key tool used in the demonstration of the effectiveness of the proposed SoS acquisition process as compared to that of the current SoS acquisition process. A reference model representing the current SoS acquisition process is built with ExtendSim modeling software with reference to the DoD SE Guide for SoS Acquisition (ODUSD(A&T)SSE, 2008) and the exploratory model proposed by DeLaurentis and Mane (2009).

Similarly, the proposed SoS acquisition model is constructed using ExtendSim. The key metric used for comparing the effectiveness of the two models is the time taken from making the SoS acquisition decision to the transition to operations and deployment from the SoS program office to the user. The times taken for both models are then compared.

C. BENEFITS OF RESEARCH

It is envisaged that the proposed SoS acquisition process can be used for future SoS development and future research work in improving the probability of success of SoS development programs.

Program managers, systems engineers, and researchers involved in SoS acquisition can benefit from this research as it provides an alternative to the current SoS acquisition process. Defense acquisition organizations can use the research outcome to effect possible enhancements to their SoS acquisition process.

D. THESIS STRUCTURE

This thesis is divided into seven chapters. The first chapter provides background understanding of the study and the problems currently faced in SoS acquisition. Chapters II and III describe the current and proposed SoS SE processes, respectively. Chapter IV

outlines the construction of the M&S models and the input parameters used, while Chapter V presents the results and provides comparison, discussions, and analysis of the results and correlates the observations with the models described in Chapters II and III.

II. CURRENT SYSTEMS ENGINEERING PROCESS IN SYSTEM-OF-SYSTEMS ACQUISITION

A. BACKGROUND

The current SoS SE process is mostly inherited from those for systems. Links between key stakeholders such as the user, SoS program office, and system program offices can be described as weak, and the program offices are almost running like silos. Inadequate coordination, span-of-control, and communications among individual systems and the SoS program offices could be one of the key reasons for the many failures in recent SoS acquisition programs. There is also a severe lack of front-end planning and SoS architecting by the SoS program office that could have led to many interoperability and compatibility issues at the verification and validation stages. These factors have resulted in the acquiring of an SoS capability at a higher development cost coupled with unacceptable delays in the delivery of capabilities and possibly inferior performance compared with what would be expected (Huynh *et al.*, 2010).

B. SYSML DIAGRAMS

As described in Chapter I, three types of SysML behavioral constructs—use case diagrams, activity diagrams, and sequence diagrams—are used to describe and present, respectively, the activities, message sequence, and high-level functionality of the current SoS SE process.

1. SysML Use Case Diagram

The SysML use case diagram in Figure 1 shows the various stakeholders (players) in the current SoS SE process. Four key groups of players are identified—decision authority, user, SoS program office, and system program offices. Their relationships and interactions in the current SoS SE process are also shown in Figure 1. As depicted in Figure 1, four players are working almost in complete silo, having very little interaction with each other. The decision authority makes the acquisition decision based on

capability needs and gaps, budget and schedule. The user determines the needs and capabilities required from the SoS. After the needs and capabilities are defined, the SoS program office will take over the translating of requirements and relaying them to the system program offices to execute the changes/modifications required. Once the systems are modified and produced, the SoS program office will oversee the integration, test and validation of the SoS. When the SoS passes all the required tests and validation, the SoS program office transfers the SoS to the user for operations and deployment.

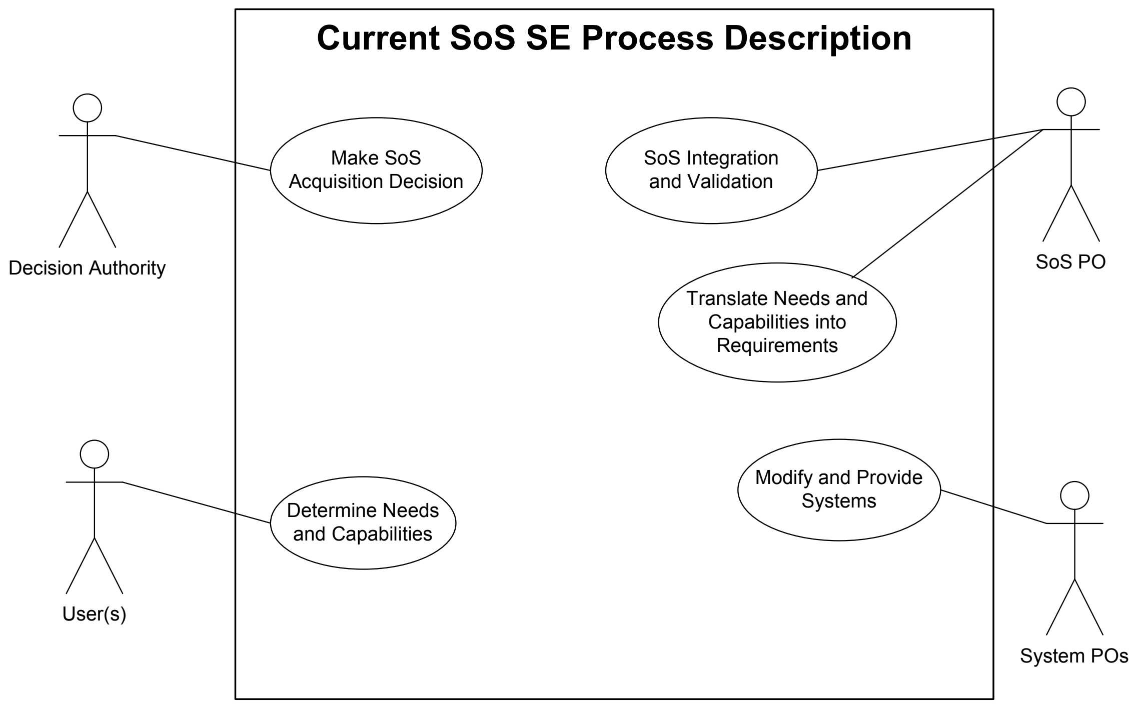

Figure 1: Use Case Diagram for Current SoS SE Process

2. SysML Activity Diagram

Figure 2 shows the SysML activity diagram that represents the current SoS SE process used by DoD programs. Swim lanes are used to represent the four key groups of stakeholders in a typical SoS acquisition environment—decision authority, user, SoS program office, and system program offices. Each swim lane contains the activities carried out by each stakeholder, and the arrows connecting each activity represent their interactions. One key observation of the current SE process is the lack of feedback at the

requirements development stage and the lack of interaction between the SoS program office and system program offices. Feedback occurs only at the test, verification, and validation activities when systems or the SoS are unable to meet the requirement or specifications, and this happens in the later part of the acquisition. This may cause the program to incur unnecessary cost and delays, resulting in the failure of the entire SoS program. The subsequent paragraphs provide a brief description of the current SE process for SoS acquisition.

The decision authority, based on requests for capability buildup from the user, approves the decision to develop the SoS capability based on a project schedule and budget. This acquisition decision is then passed down to the user and the SoS program office. Upon receiving the acquisition decision, the user proceeds to define the needs and capabilities required of the SoS based on current and projected operational needs. It is common that these needs may be inadequately defined, resulting in poor requirements being developed by the SoS program office.

The SoS program office receives the acquisition decision from the decision authority and the needs and capabilities document from the user. The SoS program office, with its pool of systems engineers and technical staff, translate the needs and capabilities into SoS requirements. This set of SoS requirements is provided to the individual system program offices. The system program offices are responsible for implementing changes and/or modifications to their respective systems that they are developing.

At each system program office, the systems engineers interpret the requirements issued by the SoS program office and issue plans and work orders to implement changes and/or modifications to their system. The amount of changes and/or modifications required may depend on the stage of development of the system and its relative complexity. A system that is nearing its completion will have a higher likelihood of more changes/modifications than one that is just starting its development. This is because the design of a system nearing completion will be more or less fixed and any addition of requirements will lead to numerous changes and/or modifications. A system that is just starting its development will experience fewer changes as its design is still evolving.

Once the changes and/or modifications to the systems are completed and satisfy the SoS requirements, the verification of the systems' ability to satisfy the SoS requirements are carried out. This ensures the system meets the SoS requirements before the actual integration of the SoS. In the event any system fails the verification, the responsible systems engineers have to review the implemented changes and/or modifications performed on the system. Remedial actions will need to be devised to meet the SoS requirements. Upon successful completion, the system program offices will issue a production "go-ahead" for the systems to be produced and sent for SoS integration.

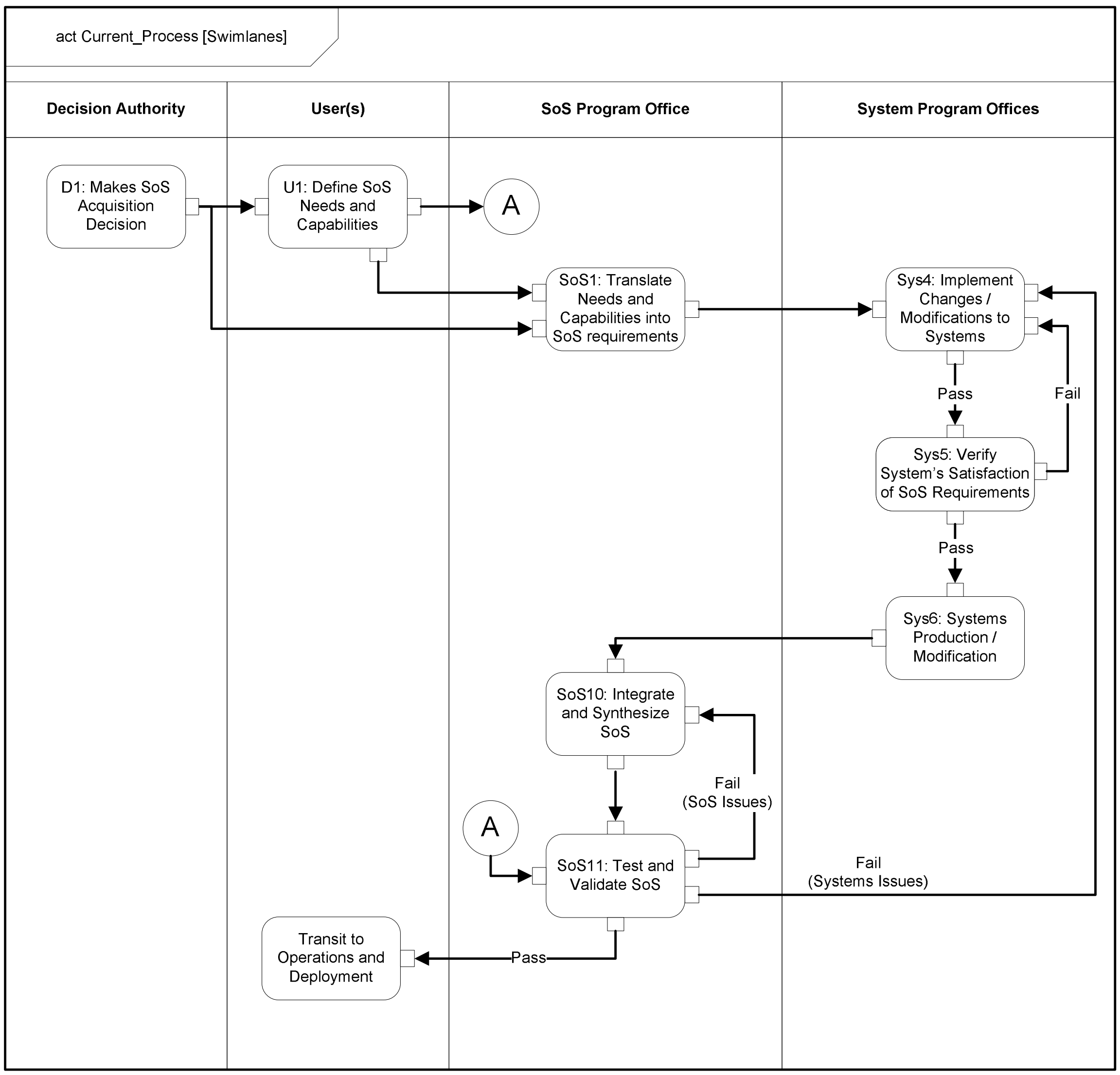

Figure 2: SysML Activity Diagram for Current SE Process for SoS Acquisition

Once the individual systems are produced, the SoS program office begins synthesizing and integrating the systems to form the SoS. This involves building interfaces and conducting interoperability checks between systems and the entire SoS. The SoS will then be tested and validated based on the needs and capabilities defined by the user. Any shortcomings will need to be fixed by reviewing the “integrate and synthesize” activity (for SoS-related issues) or implementing the change/modification activity (for system-related issues) before the SoS transits to operations and deployment and is handed over to the user.

3. SysML Sequence Diagrams

SysML sequence diagrams are used to show the messages and information passed between the stakeholders and activities. Figure 3 shows the exchange of data and information at the top level (Level 0) of the current SoS SE process.

Figures 4 through 7 show the SysML sequence diagrams for individual stakeholders (Level 1). The transfer of messages and information within the activities and with outside stakeholders are illustrated in the sequence diagrams. External stakeholders are identified by “<<ext>>” while activities performed by the stakeholders concerned are shaded. When a particular activity cannot be successfully completed, a feedback message is sent to other activities and/or stakeholders for remedial actions. Feedback messages in the sequence diagrams are represented by an asterisk (*) in front of the message description.

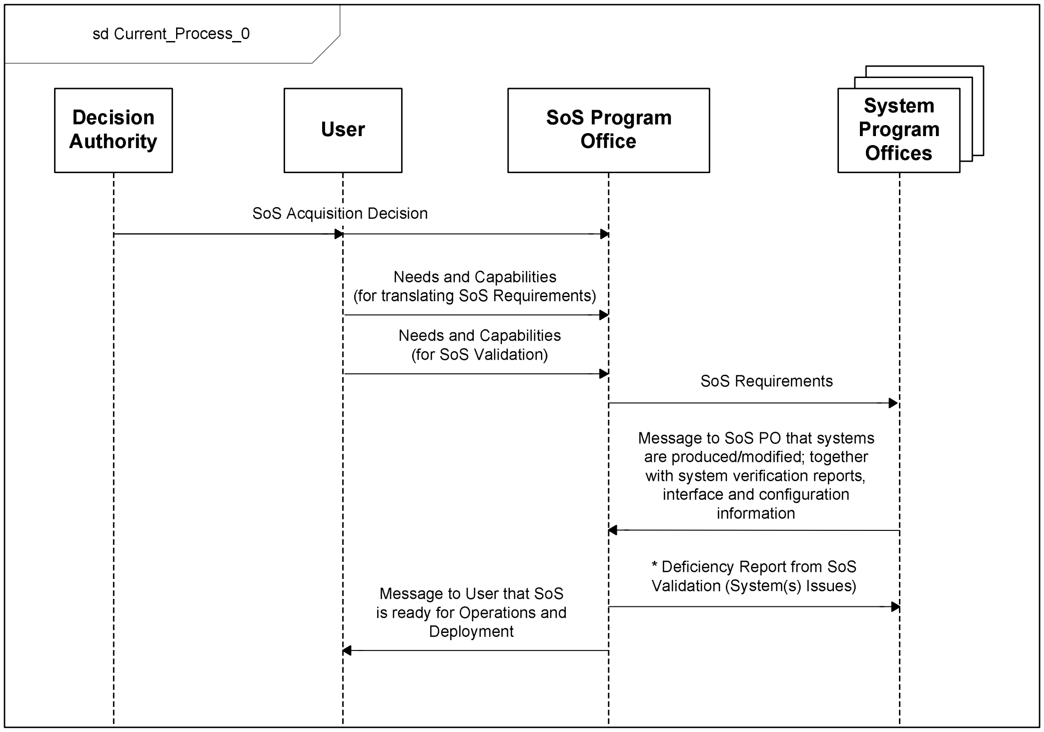

Figure 3: SysML Sequence Diagram for Current SoS SE Process (Level 0 - Overview)

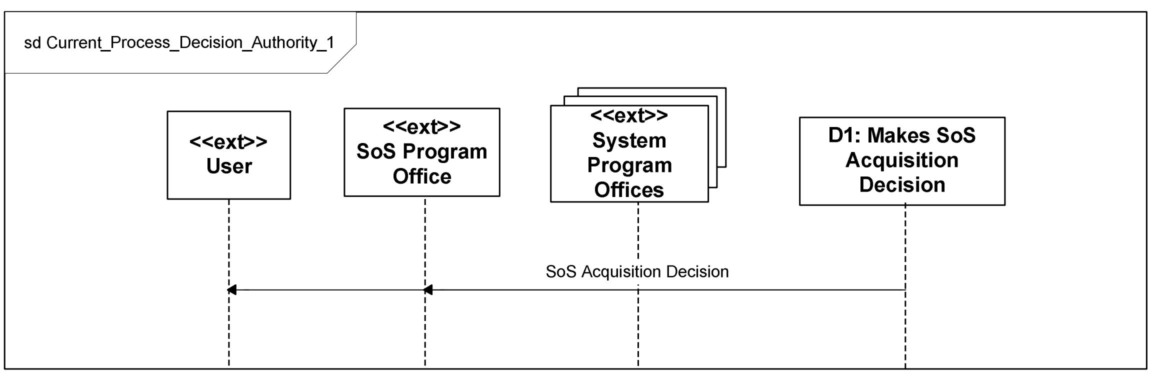

Figure 4: SysML Sequence Diagram for Current SoS SE Process (Level 1 – Decision Authority)

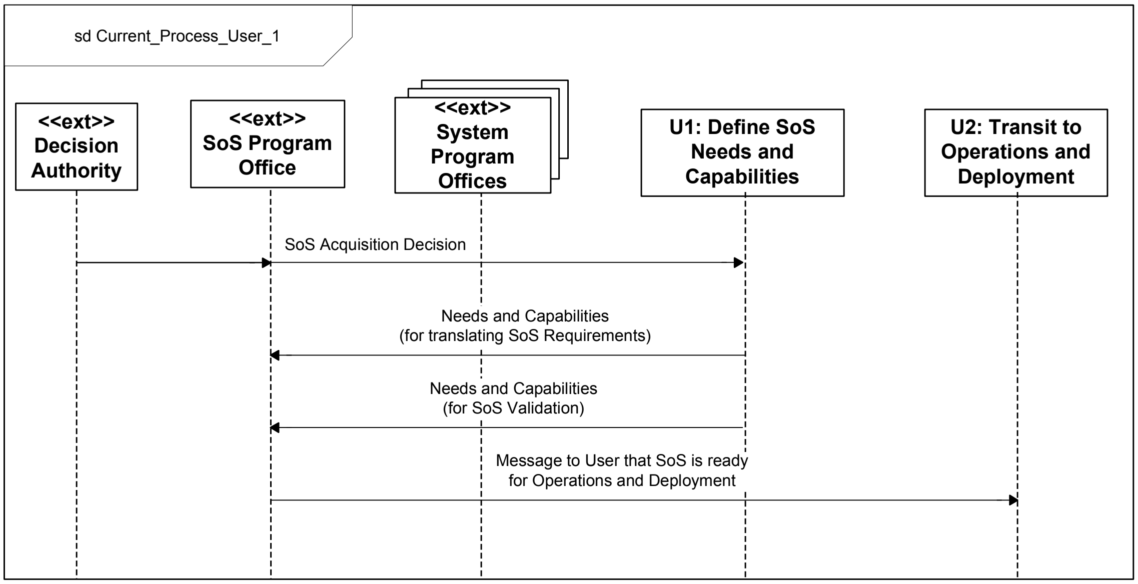

Figure 5: SysML Sequence Diagram for Current SoS SE Process (Level 1 – User)

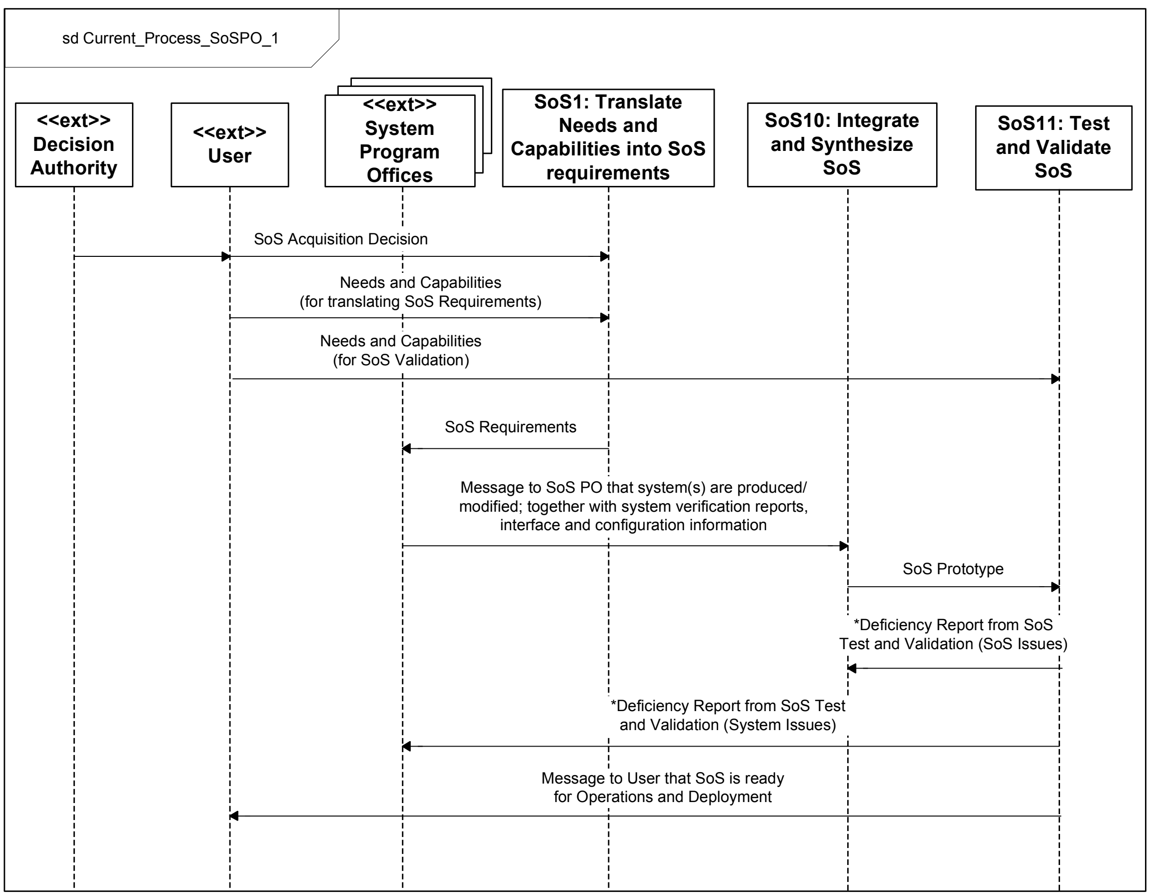

Figure 6: SysML Sequence Diagram for Current SoS SE Process (Level 1 – SoS Program Office)

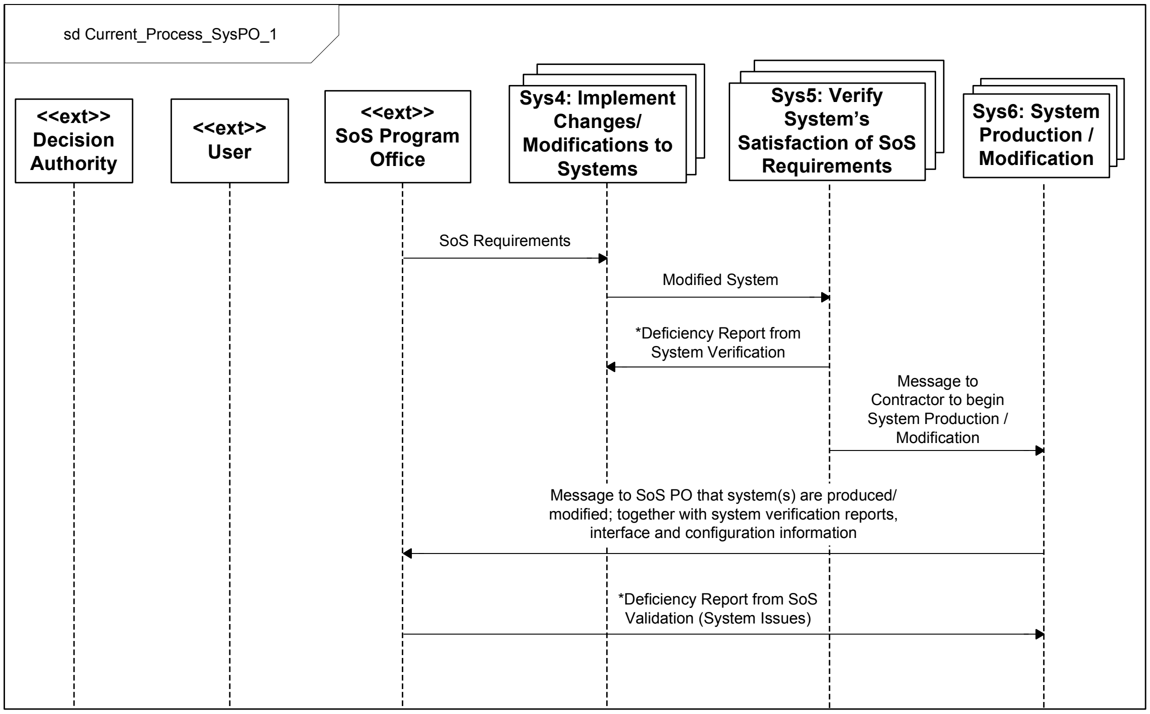

Figure 7: SysML Diagram for System Program Offices (Level 1) for Current SE Process

III. PROPOSED SYSTEMS ENGINEERING PROCESSES IN SYSTEM-OF-SYSTEMS ACQUISITION

A. BACKGROUND

A set of front-end SE activities is proposed for addition to the current SE process for SoS acquisition to increase interaction between the various stakeholders. With the proposed SoS SE process, the SoS program office has a wider span-of-control over the development of the SoS through the development of the SoS architecture in consultation with the individual system program offices. Constant feedback from the SoS activities would allow shortcomings to be addressed early and could help to reduce the chance of failure in the later stages of the acquisition process.

B. SYSML DIAGRAMS

As discussed in Chapter II, SysML diagrams (use case, activity, and sequence diagrams) are used to represent, respectively, the activities, message sequence, and high-level functionality of the proposed SoS SE process.

1. SysML Use Case Diagram

Figure 8 shows the use case diagram for the proposed SoS SE process. An additional function (SoS architectural development) is included to reflect the front-end SE activities that will be performed before the actual development of the SoS begins. This increases the interaction between the SoS program office and the individual system program offices. The SoS architecture, which is developed in consultation with the individual system program offices, forms the basis from which the individual system program offices perform changes and modifications to their systems to meet the SoS requirement. In addition, the user is now linked to the "Translate Needs and Capabilities into Requirements" with the SoS program office. This can be achieved by having the SoS program office conduct verification and validation of SoS requirements with the user before finalizing them. This two-way communication helps to address any concerns by

the SoS program office when formulating the SoS requirements and reduces the chance of misinterpreting the user's needs, which could lead to an undesirable product or one that needs massive re-work.

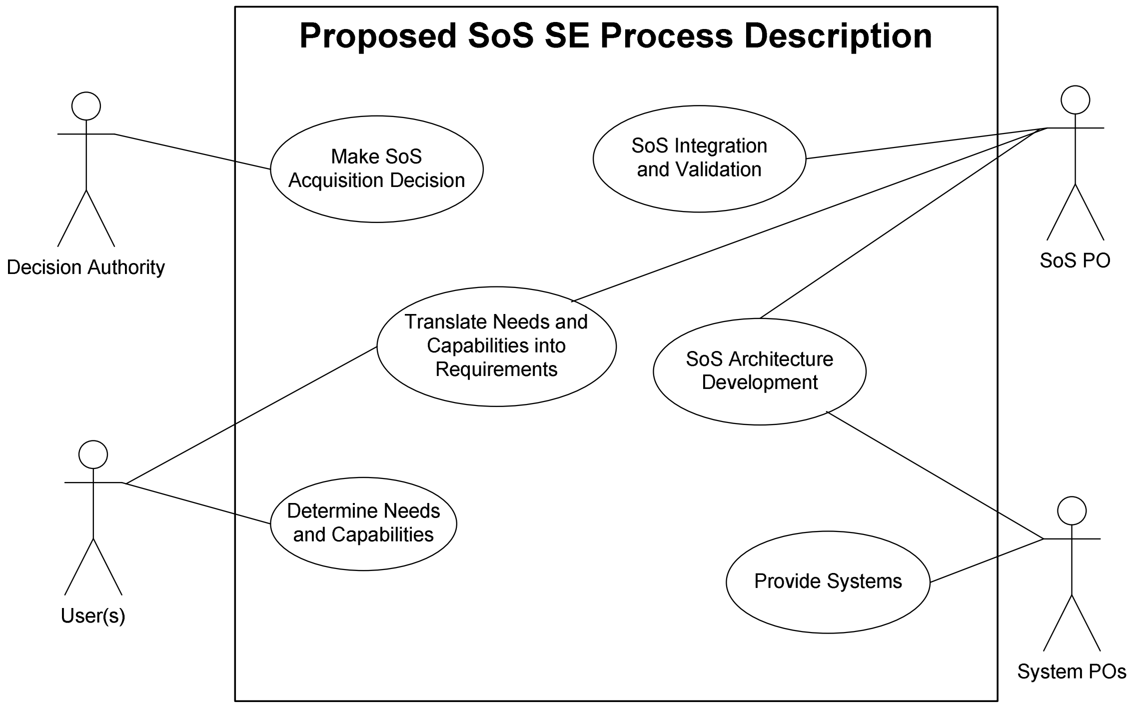

Figure 8: Use Case Diagram for Proposed SE Process for SoS Acquisition

2. SysML Activity Diagram

Figure 9 shows the SysML activity diagram for the proposed SoS SE process. One of the key features is the inclusion of a front-end SE process (shaded) that focuses on developing an over-arching SoS architecture first before the actual SoS development. While developing the SoS architecture will incur some time, it is likely that increased interactions between the SoS and individual system program offices and the systematic application of sound systems engineering principles will lead to an increase in probability of success for each activity and the program.

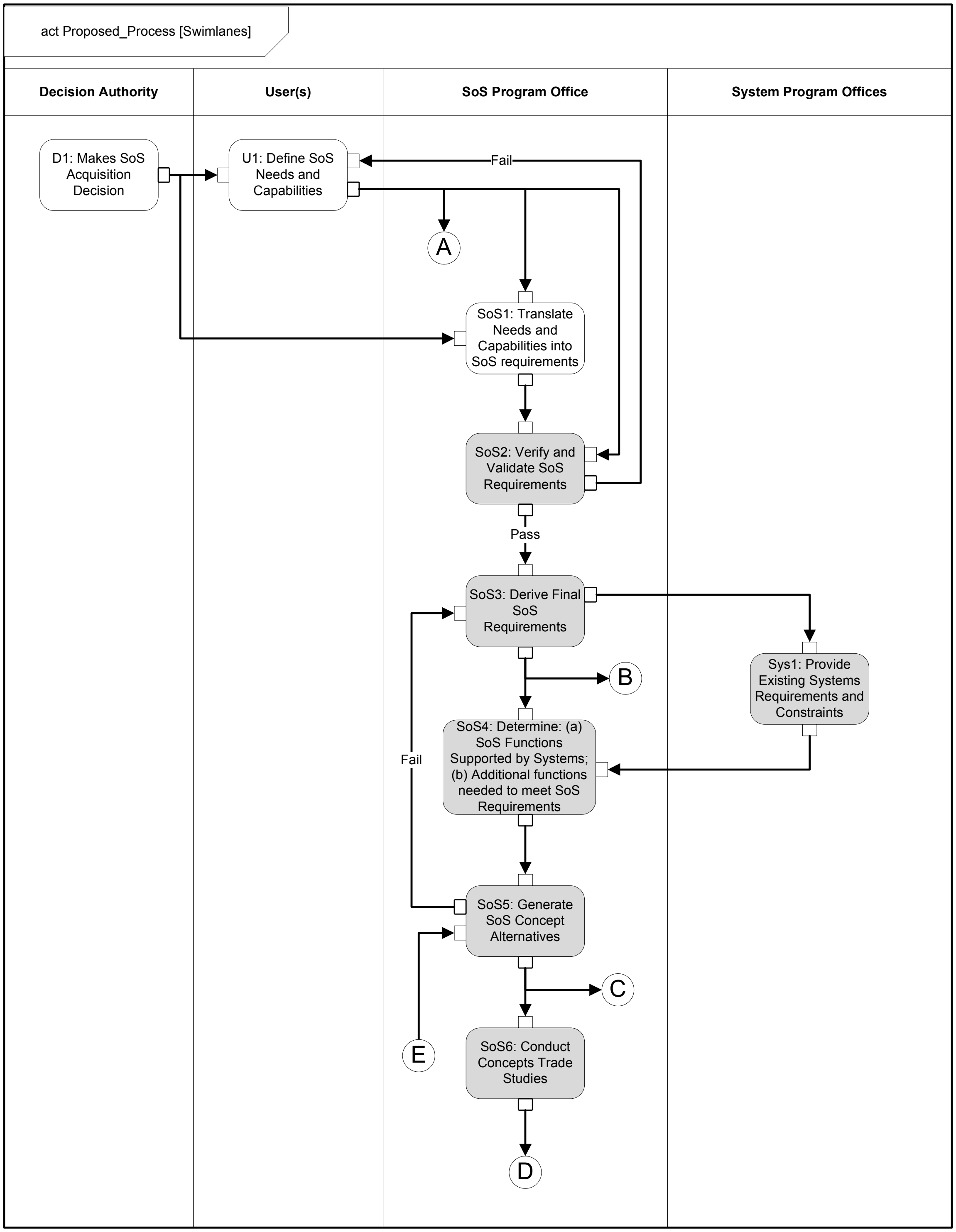

Figure 9 (a): SysML Activity Diagram for Proposed SE Process for SoS Acquisition

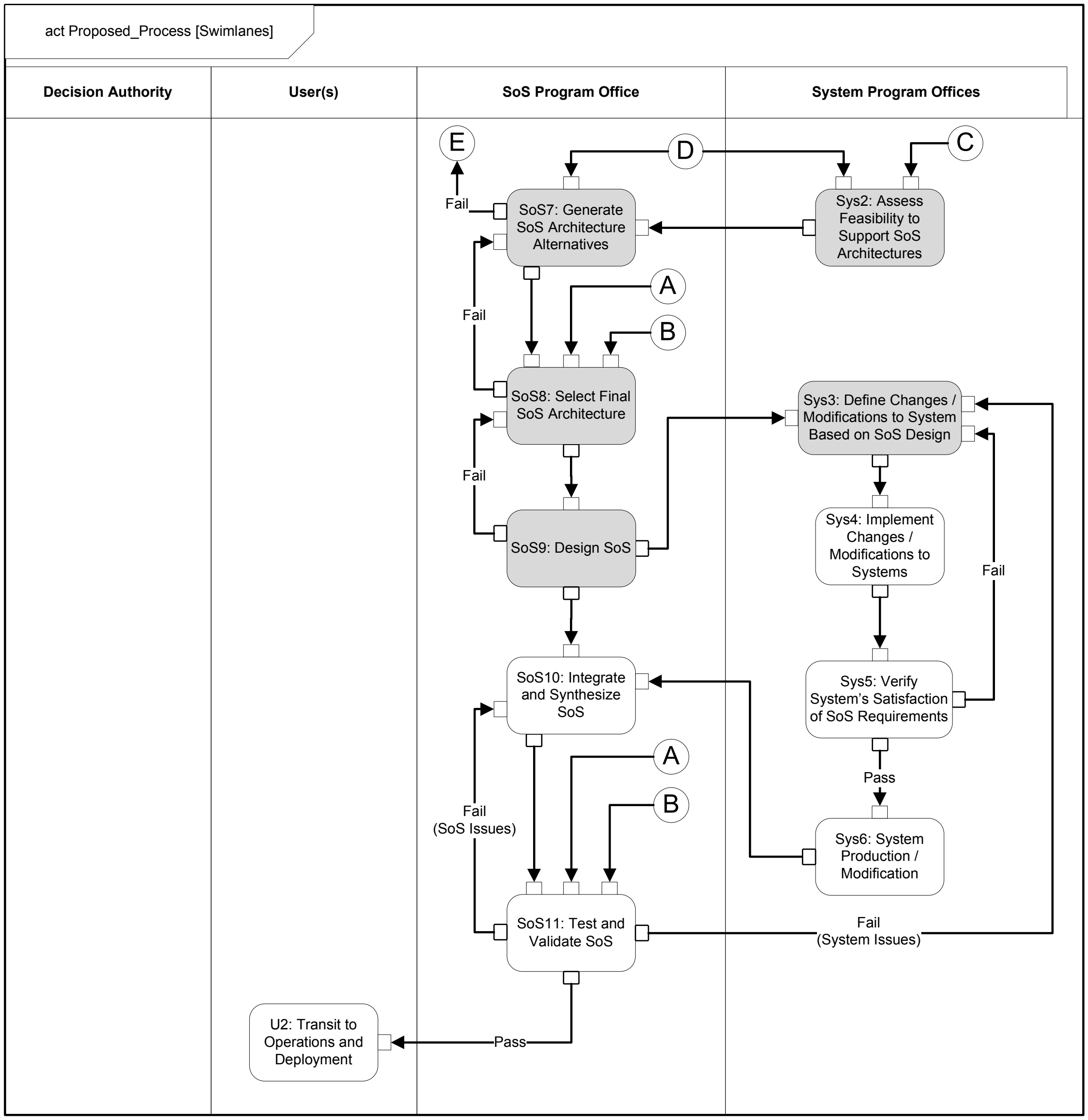

Figure 9(b): SysML Activity Diagram for Proposed SE Process for SoS Acquisition

Figure 9(a) indicates that, having translated the needs and capabilities into SoS requirements, the SoS program office verifies and validates these SoS requirements with the user to ensure that the SoS requirements correctly represent the user needs and are feasible to implement and testable. This activity prevents the possibility of inaccurate or incorrect requirements being passed down to the activities in the later stages. Once the SoS requirements are verified and validated with the user, the SoS program office derives the final SoS requirements. The final SoS requirements are then sent to the individual

system program offices, and the SoS program office requests the respective system specifications and constraints. These specifications and constraints, together with the final SoS requirements, serve as important inputs to the SoS program office in determining which of the SoS functions can be fulfilled by the systems and which SoS function(s) are needed to meet the final SoS requirements. The final SoS requirements give the system program offices a preview to the tasks ahead and help them in determining the constraints of their systems in supporting the SoS.

Functional allocation is conducted to determine which functions of the SoS are not met by the systems and need to be developed separately. After the functions are appropriately allocated, the SoS program office starts generating the SoS concept alternatives that could meet the SoS requirements derived earlier. These concept alternatives are subjects of concept trade studies. The concept alternatives, together with the outputs of the concept trade studies, are also sent to the system program offices as inputs for their assessment on their feasibility to support the various SoS architectural alternatives proposed by the SoS program office.

The outputs of the concept trade studies by the SoS program office and the assessment of the ability of the individual system program offices to meet the SoS architecture requirements are used as inputs to generate SoS architectural alternatives. If there are problems in developing the SoS architecture alternatives, the SoS program office will review the concept alternatives. The final SoS architecture is selected with reference to the needs and capabilities defined by the user in the early stage of the acquisition process and the final SoS requirements derived in SoS3 by the SoS program office.

3. SysML Sequence Diagram

As in Chapter II, SysML sequence diagrams are used to show the messages and data passed between the various stakeholders and activities in the proposed SoS SE process. The activities in grey blocks shown in Figures 12 and 13 are unique to the proposed SoS SE process. Figure 10 shows the exchange of data and messages at the top level of the SoS acquisition (Level 0) using the proposed SoS SE process.

Figures 11 through 13 show the next level of detail (Level 1) of the SysML sequence diagram. These sequence diagrams show the message and data transfer between activities and external stakeholders. The external stakeholders are identified by "<<ext>>". When a particular activity cannot be successfully completed, a feedback message is sent for remedial actions. Feedback messages in the sequence diagrams are represented by an asterisk (*) in front of the message description.

One key observation from the SysML sequence diagrams for the current and proposed SE process is the increased number of messages sent to and from the SoS program office and the system program offices. The number of messages between the SoS program office and the user at the start of the SoS acquisition also increases. This increase in communications and span-of-control by the SoS program office is envisaged to improve the probability of success of the program (as mentioned in Chapter I).

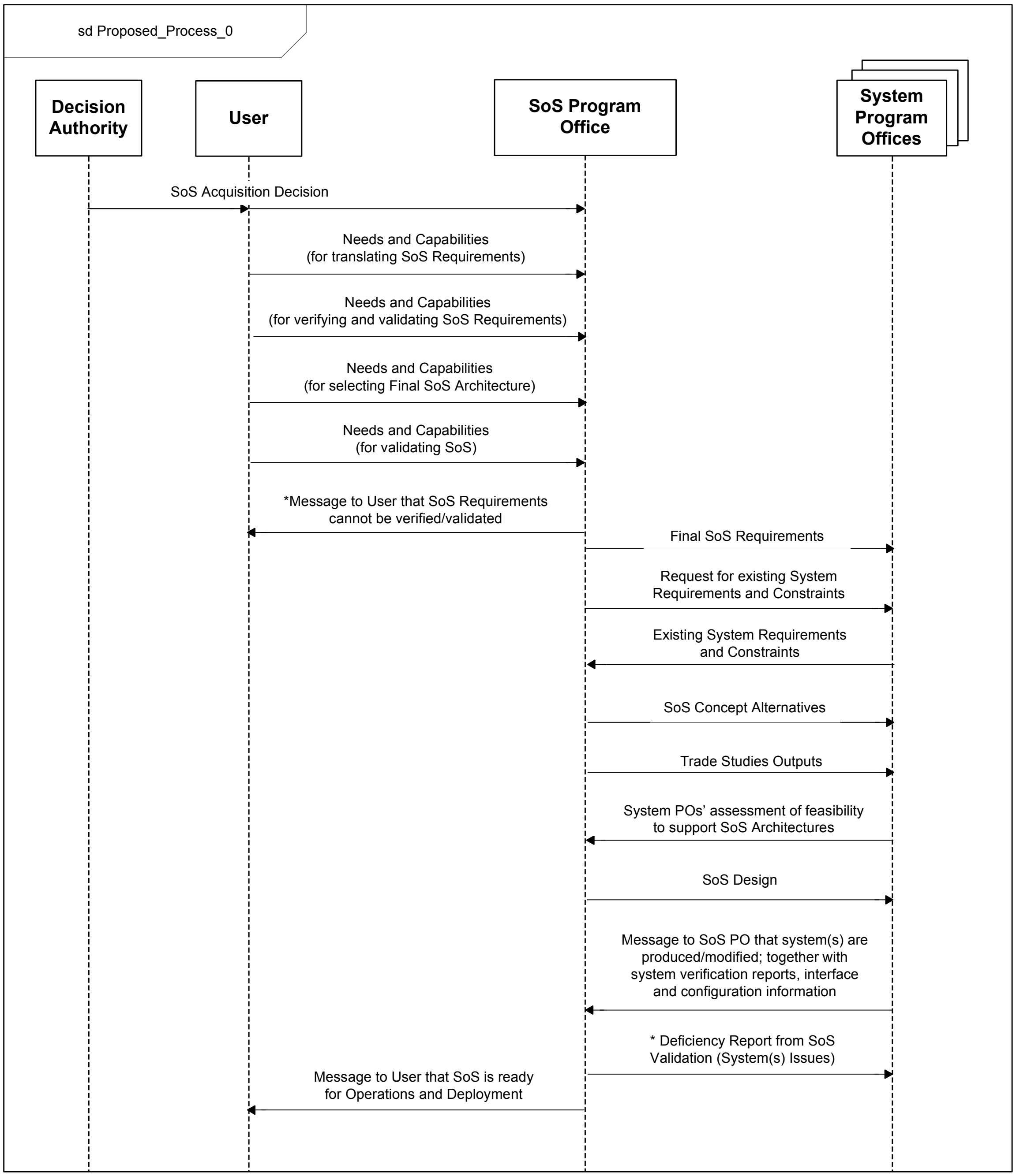

Figure 10: SysML Sequence Diagram for Proposed SoS SE Process (Level 0)

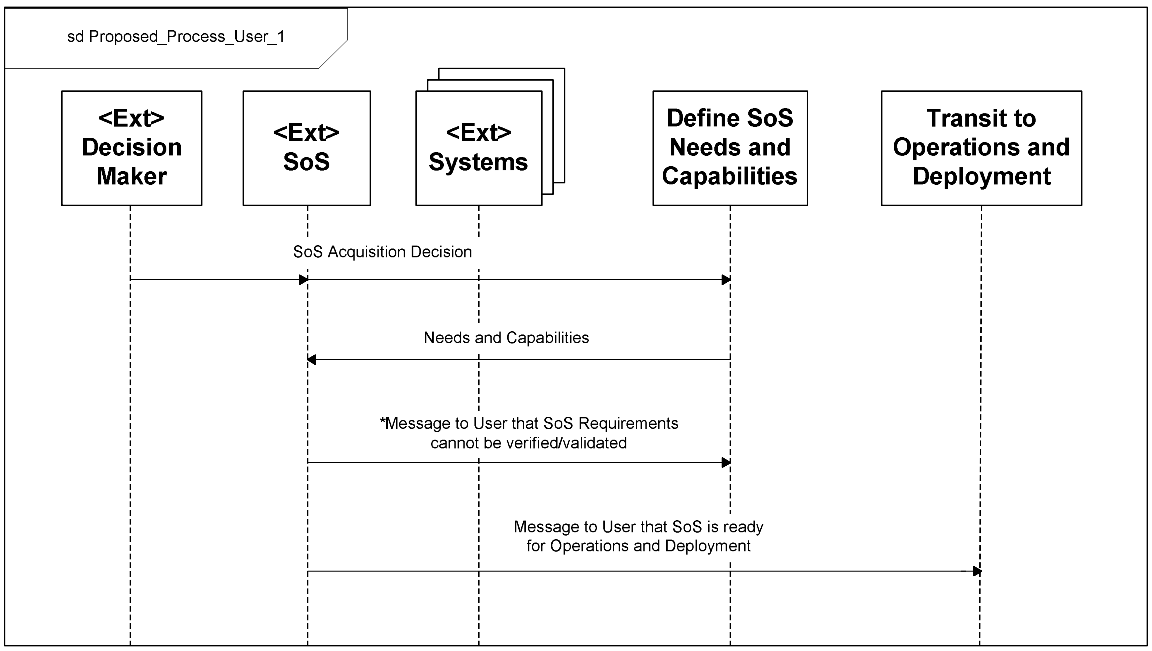

Figure 11: SysML Sequence Diagram for User (Level 1) for Proposed SoS SE Process

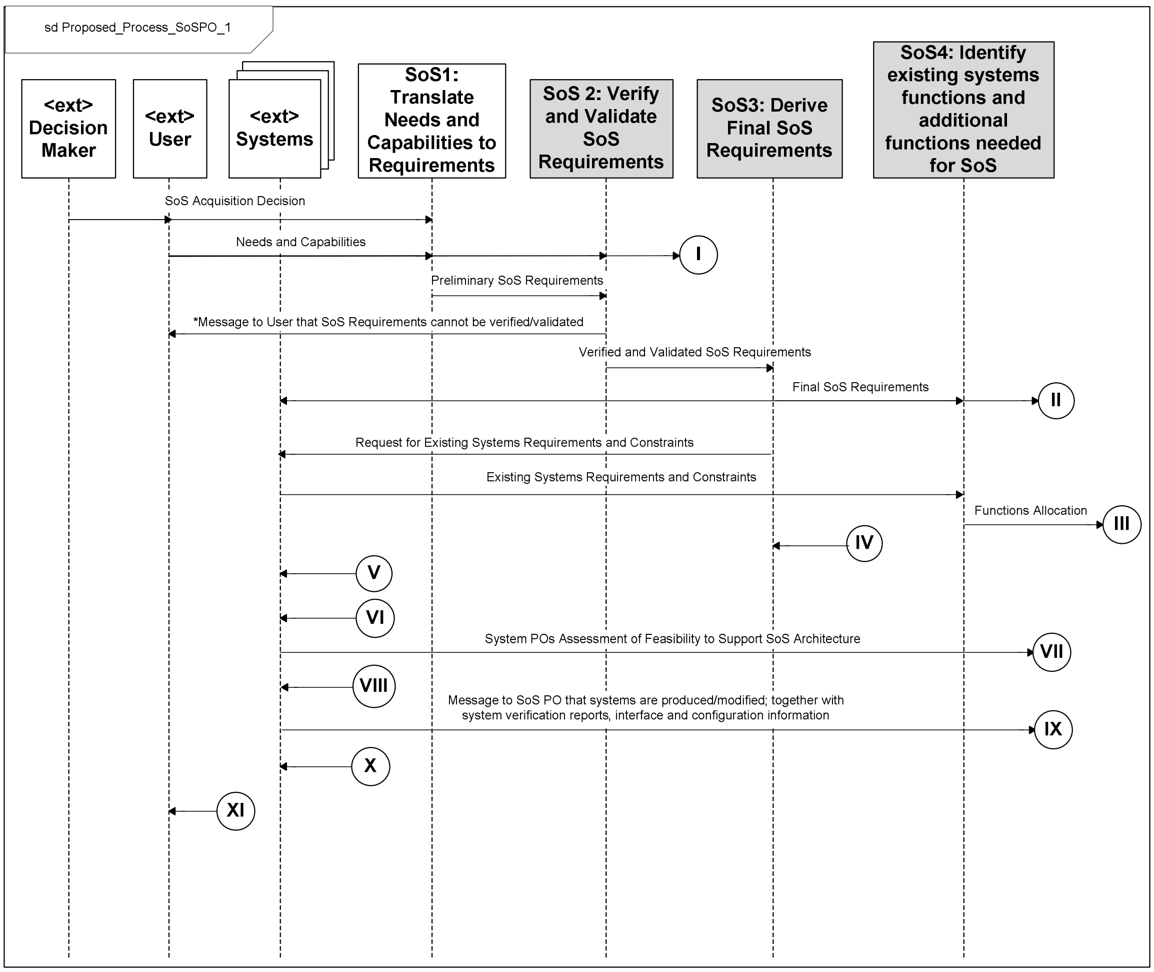

Figure 12(a): SysML Sequence Diagram for SoS Program Office (Level 1) for Proposed SoS SE Process

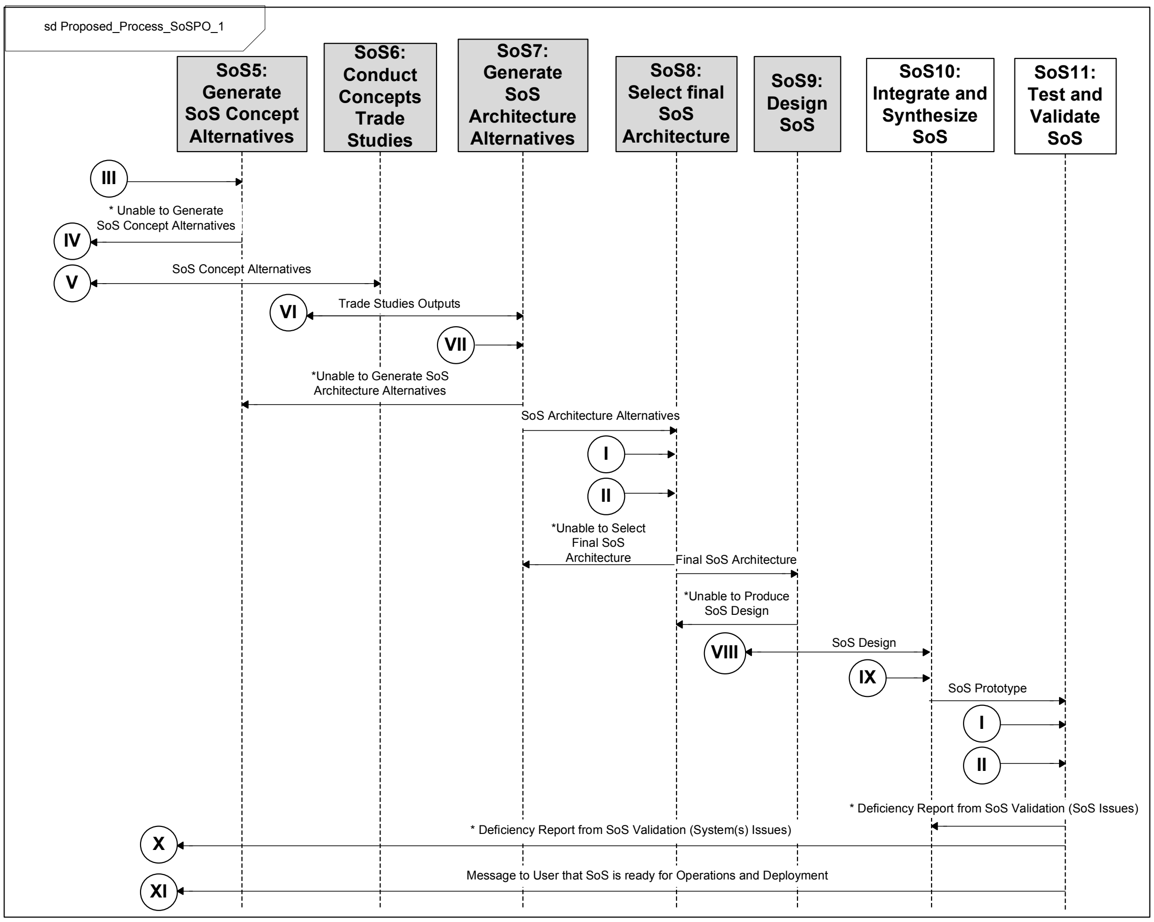

Figure 12(b): SysML Sequence Diagram for SoS Program Office (Level 1) for Proposed SoS SE Process

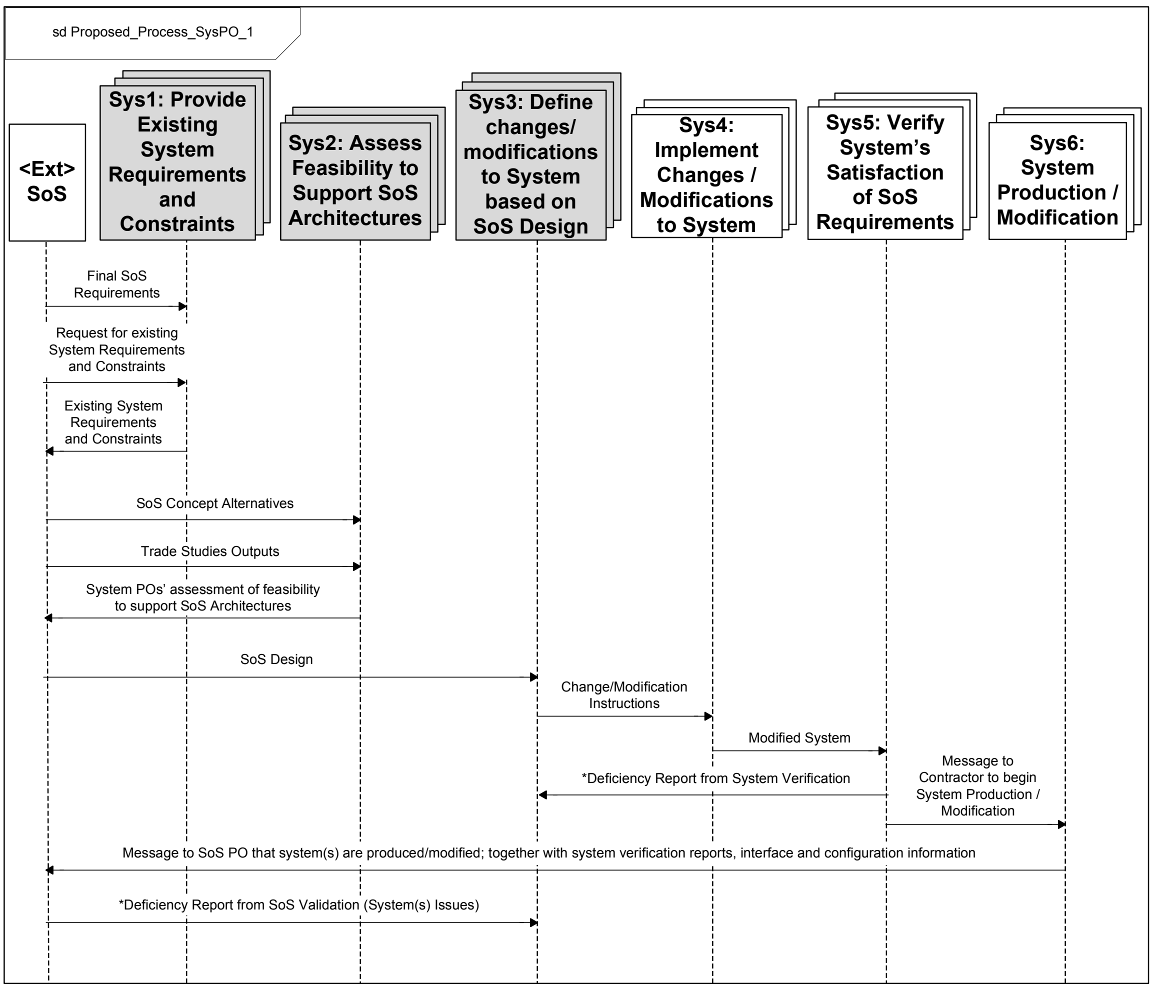

Figure 13: SysML Sequence Diagram for System Program Offices (Level 1) for Proposed SoS SE Process

THIS PAGE INTENTIONALLY LEFT BLANK

IV. COMPARATIVE ANALYSIS

A. EXTENDSIM MODEL FOR CURRENT SOS SE PROCESS

1. Hierarchical Modeling

The ExtendSim model of the current SoS SE process is structured into two levels. The first level (known as Level 0) consists of the top-level view of the current SoS SE process that comprises the four key players—decision authority, user, SoS program office, and system program offices. Figure 14 shows the top-level view (Level 0) of the current SoS SE process model implemented on ExtendSim.

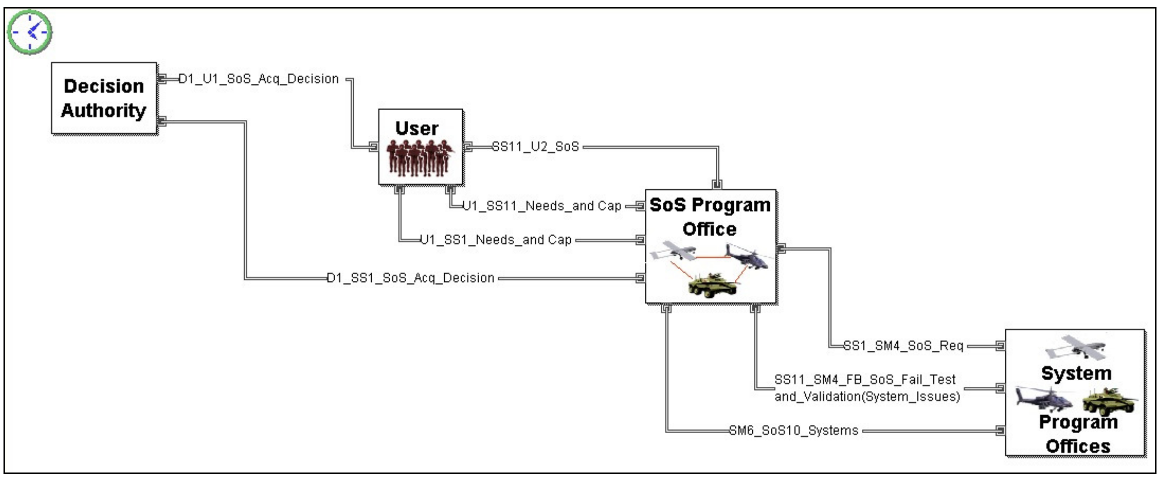

Figure 14: Overview of ExtendSim Model: Current SoS SE Process Model (Level 0)

Unlike the decision authority, the other three players (user, SoS program office and system program offices) perform more than one activity. The ExtendSim models for the activities of each player are shown in Figures 15 through 17. These are known as the Level 1 (or stakeholder level) view of the current SoS SE process model. The next level (Level 2) comprises detailed modeling using ExtendSim blocks and functions. Level 2 details are shown in Appendix A.

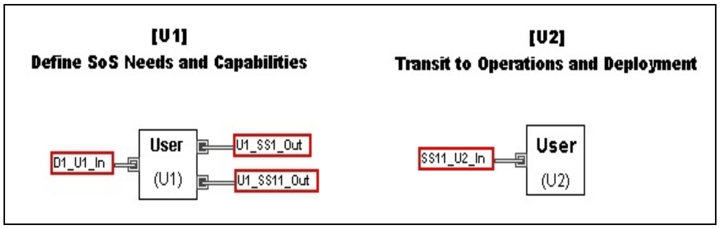

Figure 15: ExtendSim Model for Current SoS SE Process (Level 1 – User)

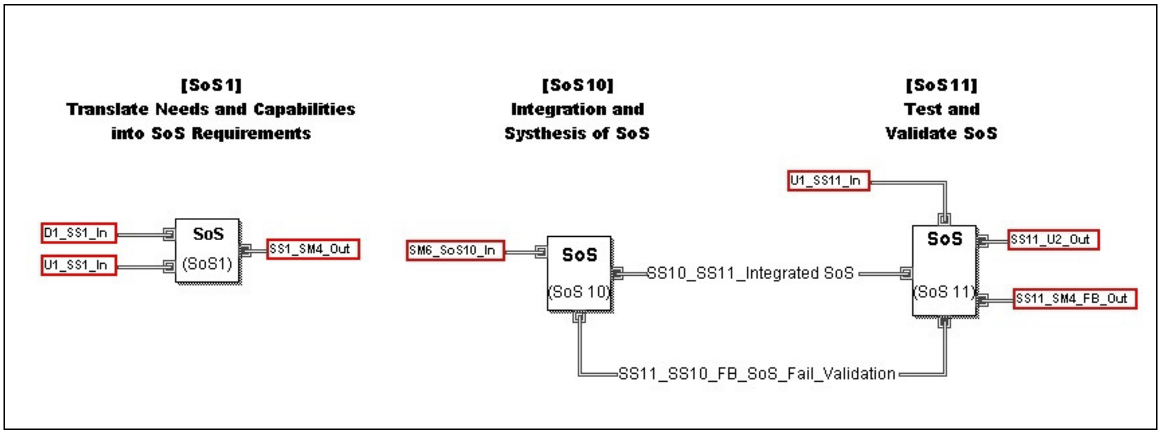

Figure 16: ExtendSim Model for Current SoS SE Process (Level 1 – SoS Program Office)

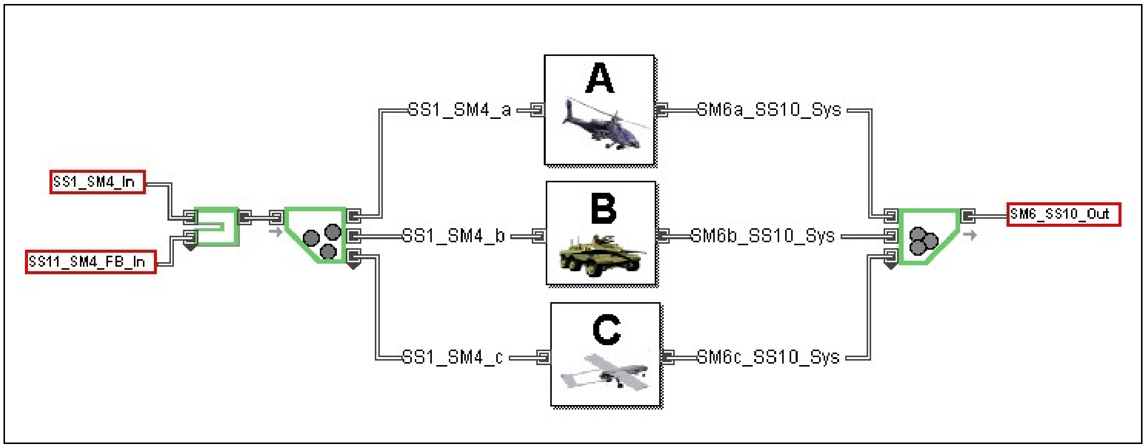

Figure 17: ExtendSim Model for Current SoS SE Process (Level 1 – System Program Offices)

2. Model Dynamics for Current SoS SE Process

The current SoS SE process uses the Discrete Event Simulation in ExtendSim to simulate the flow of information, data, and/or hardware among the four players and through the various activities. The flow of the model reflects the SysML activity diagram described in Chapter II. A Monte Carlo simulation using 1000 runs is performed on the ExtendSim model in order to obtain the average processing time for each activity and total time required to complete the acquisition process.

The "Create" block in ExtendSim simulates an acquisition decision made by the decision authority by releasing an item into the model. This item represents the flow of messages, information, or hardware as described in the SysML sequence diagrams in Chapter II. The item passes through several activity blocks representing the activities in the SE process. The time taken to complete an activity is approximated by ExtendSim by predetermining a probability density function with a mean processing time. Also, along the way, the item may encounter feedback loops that require it to perform an activity(s) again. These feedback loops simulate the failure or unsuccessful completion of an activity, which requires the program office to review or provide remedies. The probability of feedback occurring is simulated by the program drawing a random number and comparing it with the probabilities that are provided to the model. The probability of failure of selected activities is discussed in Section C. The item eventually exits the simulation modeling. This represents the completion of the SoS development and the transition to operations and deployment.

B. EXTENDSIM MODEL FOR PROPOSED SE PROCESS

1. Hierarchical Modeling

The structure of the ExtendSim model for the proposed SoS SE process is similar to that for the current SoS SE process (Figures 14 through 17). The top level (Level 0) is still made up of the four key stakeholders of the SoS acquisition. The key differences between the current and proposed SoS SE processes are that there are more communications and interactions among the user, SoS program office, and system

program offices (Figure 18), and more activities within the SoS program office and system program offices (Figure 19 through 21). Details of Level 2 modeling are in Appendix B.

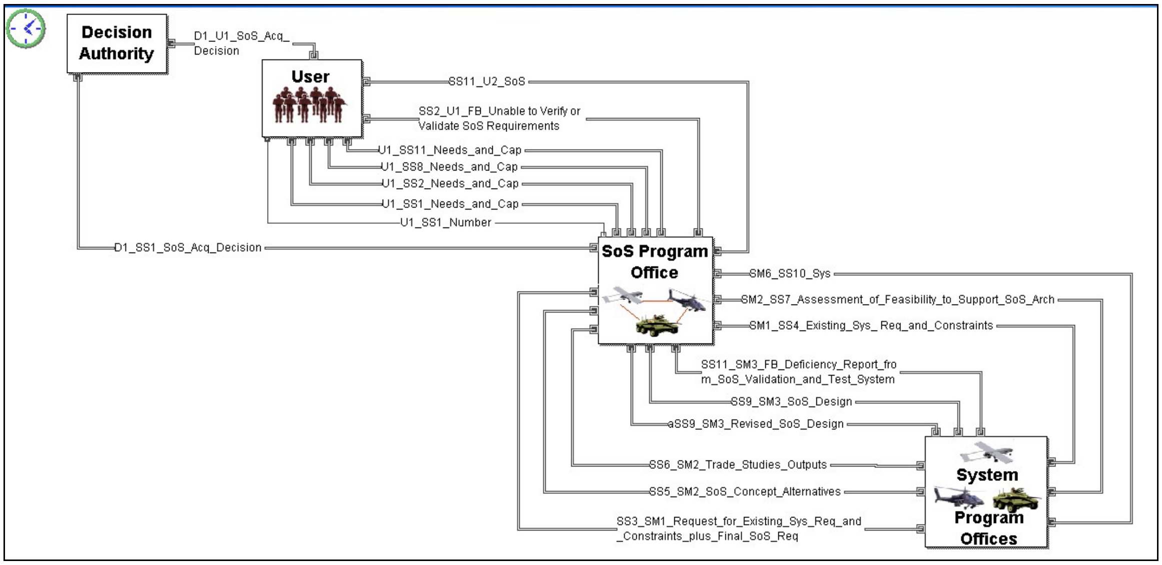

Figure 18: ExtendSim Model for Proposed SoS SE Process (Level 0 – Overview)

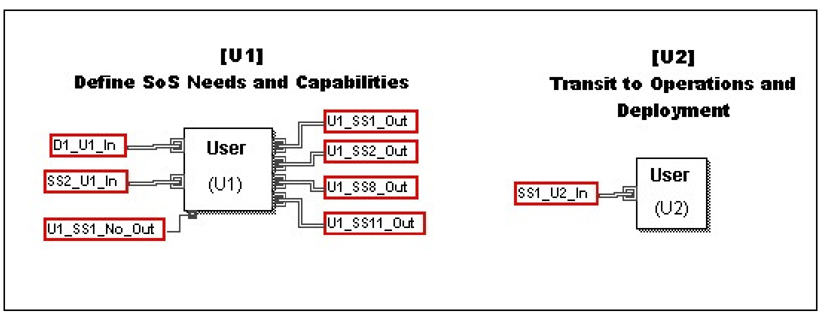

Figure 19: ExtendSim Model for Proposed SoS SE Process (Level 1 – SoS Program Office)

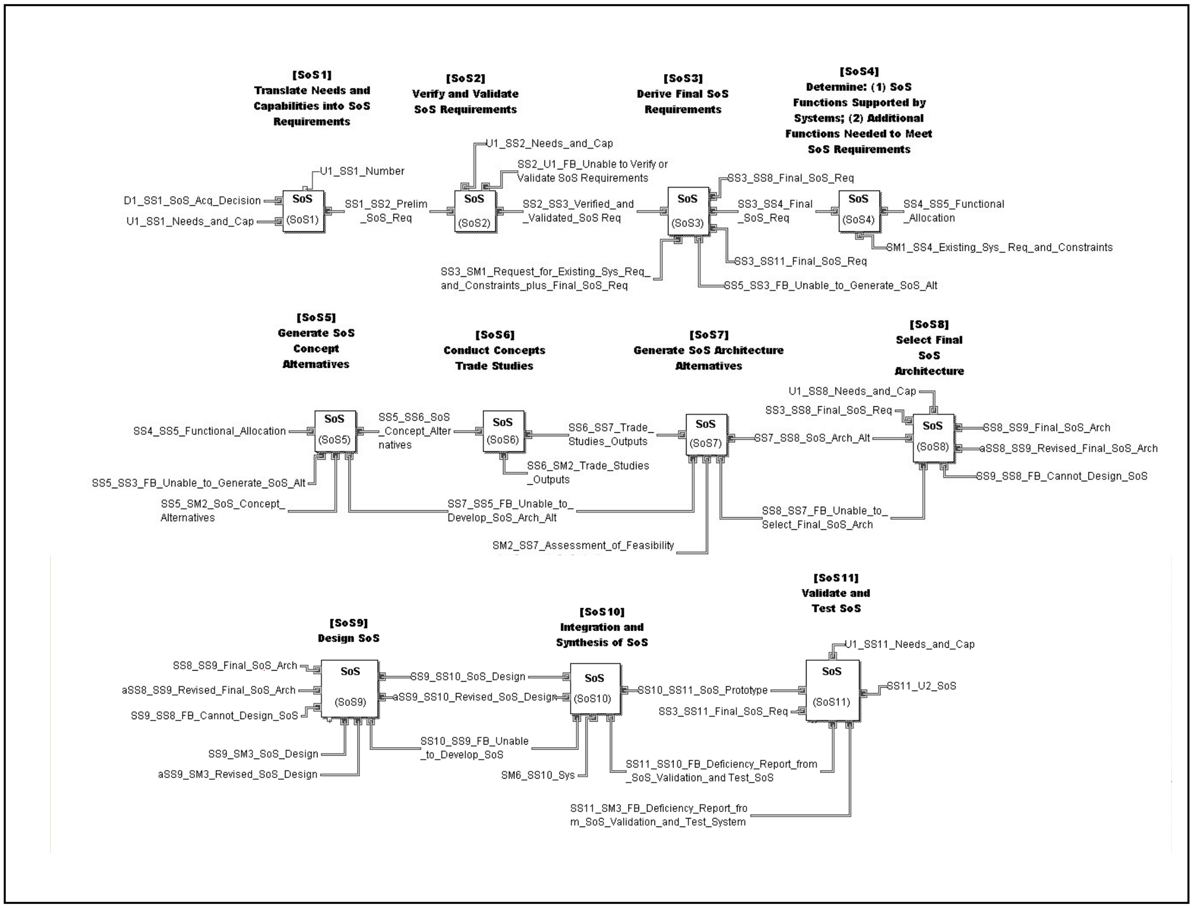

Figure 20: ExtendSim Model for Proposed SoS SE Process (Level 1 – SoS Program Office)

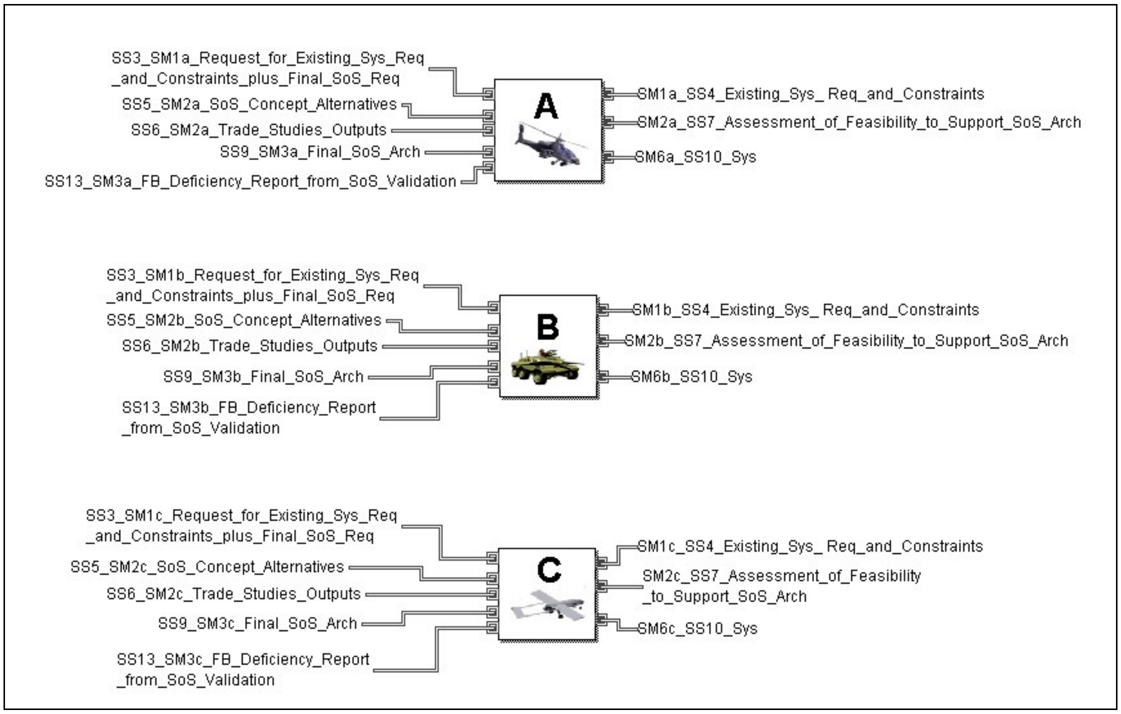

Figure 21: ExtendSim Model for Proposed SoS SE Process (Level 1 – System Program Offices)

2. Model Dynamics for Proposed SoS SE Process

The modeling and simulation of the proposed SoS SE process also uses ExtendSim to simulate the flow of information, data, and/or hardware among the four players and through the various activities. Furthermore, a Monte Carlo simulation using 1000 runs is also used to obtain the average processing time for each activity and total time required to complete the acquisition process. The flow of the simulation model reflects the SysML activity diagram described in Chapter III.

C. INPUT PARAMETERS

1. Mean Processing Time

The mean processing time is one of the key inputs of the ExtendSim program. It is used in the determination of the amount of time (in time units) required at each activity

block. It is determined by an ExtendSim random block that generates a delay input to the activity block using the mean and distribution provided. The Beta distribution is selected to approximate the time required to complete an activity, as this distribution is often used for task completion time in the absence of data. The shape of the density function is characterized by two shape parameters (α_1 and α_2). Based on experience with real world data, the Beta distribution is expected to skew to the right with $\alpha_2 > \alpha_1 > 1$ (Law, 2007). From Figure 22, the shape parameters α_1 =1.5 and α_2 =3.0 are selected for this research, as they give a more gradual variation than do the shape parameters α_1 =1.5 and α_2 =5.0. The Beta distribution with shape parameters α_1 =1.5 and α_2 =3.0 are used to determine the processing time required for each activity throughout the simulation work. The subsequent paragraphs will address the other parameter needed for the Beta distribution—the upper bound or maximum value of the distribution (a required input).

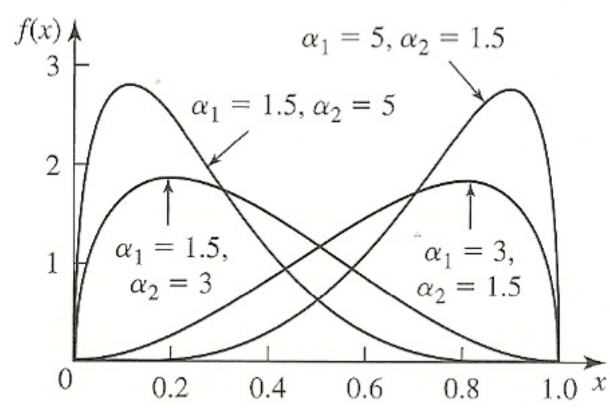

Figure 22: Probability Density Function for Beta Distribution for Different Shape Parameters (α_1 and α_2) (From Law, 2007)

The upper bound, b, of the Beta distribution governing an activity can be determined from the lower bound, a, and the mean processing time, μ, using the following expression (Law, 2007):

$$\mu = a + \frac{\alpha_1 (b-a)}{\alpha_1 + \alpha_2}. \tag{1}$$

Using α_1 =1.5, α_2 =3.0, setting the lower bound of the density function as zero, and arranging the expression above then yield

$$b = 3\mu \,. \tag{2}$$

Since the different activities may vary in complexity and magnitude, a standardized set of mean processing times is used. Three main categories of activities with significantly different processing times are identified: Analysis and Design (A&D); Testing, Verification, and Validation (TVV); and Integration, Modification, and Production (IMP).

a. Analysis and Design

This category of activities generally deals with work done on paper, and these activities are mainly the front-end activities of the SE process. A&D covers activities such as translating needs and capabilities, generating alternatives, selecting alternatives, and designing of the SoS.

b. Testing, Verification, and Validation

This category of activities involves testing of the system(s) and/or SoS and verifying and validating its requirements satisfaction. These activities generally result in a pass or fail outcome. It is reasonable to assume that, because of extensive preparation, pre-trial tests, and the trial duration, testing, verification, and validation activities take more time than do the analysis and design activities. Hence, a ratio of 1:3 (A&D:TVV) in the mean processing time is assumed in the simulation study conducted in this research.

c. Integration, Modification, and Production

This category of activities concerns the physical assembling and manufacturing of the systems and/or SoS. From an investigation of reports from the Government Accountability Office (GAO) on ten system acquisition programs as shown in Table 1, the average of the test time to production time ratios for the ten programs is obtained. Parenthetically, averaging the ratios ensures that any bias is minimized, as

different programs have different levels of complexities and development times. This average ratio of the processing time (TVV:IMP) is used as a proxy to determine the mean processing time for IMP activities for both the SoS and system program offices.

Table 1: Testing and Production Times for Ten Different System Acquisitions (Data obtained from GAO Reports)

S/N	System	Testing Duration (T)	Production Duration (P)	Ratio (T:P)
1	C-17 (GAO, 1989)	18	49	2.7
2	Navy Theater Wide Block 1 (GAO, 2000)	42	144	3.4
3	CH-53K (GAO, 2011)	84	96	1.1
4	E-10A (GAO, 2005)	15	48	3.2
5	DD(X) (GAO, 2004)	53	36	0.68
6	Joint Cruise Missile (GAO, 2010)	12	24	2.0
7	Howitzer (GAO, 2000, 2002)	23	28	1.2
8	Longbow Apache (GAO, 1991)	12	24	2
9	LHX (GAO, 1988)	15	24	1.6
10	Expeditionary Fighting Vehicle (GAO, 2010)	51	66	1.3
Sub-Total		325	539	
Ratio (Average of (T:P))				1.9

The amount of time required for each activity for different programs varies significantly. Instead of using absolute values for the mean processing time for each activity, normalized time units are used. From the analysis and observations discussed in the preceding paragraph, the time unit ratio for analysis (A), testing (T), and production/assembly (P) is 1:3:5.7. Table 2 lists the mean processing time (in time units) for each activity and is held constant for this simulation study. Note that an activity with an asterisk (*) indicates that the activity is only applicable to the proposed SE process for SoS acquisition. Also note that the start and end activities (i.e., make SoS acquisition decision and transit to operations and deployment) do not incur any processing time.

Table 2: Mean Processing Time for Each Activity

S/N	Activity	Mean Processing Time (Units)
D1	Makes SoS Acquisition Decision	0
U1	Define SoS Needs and Capabilities (A)	1
U2	Transit to Operations and Deployment	0
SoS1	Translate Needs and Capabilities into SoS Requirements (A)	1
SoS2	Verify and Validate SoS Requirements* (A)	1
SoS3	Derive Final SoS Requirements* (A)	1
SoS4	Determine: (a) SoS functions supported by Systems; (b) Additional functions needed to meet SoS requirements* (A)	1
SoS5	Generate SoS Concept Alternatives* (A)	1
SoS6	Conduct Concepts Trade Studies* (A)	1
SoS7	Generate SoS Architecture Alternatives* (A)	1
SoS8	Select Final SoS Architecture* (A)	1
SoS9	Design SoS* (A)	1
SoS10	Integrate and Synthesize SoS (P)	5.7
SoS11	Test and Validate SoS (T)	3
Sys1	Provide Existing Systems Requirements and Constraints* (A)	1
Sys2	Assess Feasibility to Support SoS Architecture* (A)	1
Sys3	Define Changes/Modifications to Systems Based on SoS Architecture and Associated Requirements* (A)	1
Sys4	Implement Changes / Modifications to Systems (P)	5.7
Sys5	Verify System's Satisfaction of SoS Requirements (T)	3
Sys6	System Production / Modification (P)	5.7

2. Probability of Success for Each Activity

The selected activities in both the current and proposed SE processes for SoS acquisition (as listed in Table 3) can fail for various practical reasons. For example, in the "Sys5: Verify system's satisfaction of SoS requirements" activity performed under the system program offices in the current SE process, there is a possibility that the system may fail the verification test. When this happens, the system program office(s) goes back to the "Sys4: Implement Changes/Modifications to System" activity for remedial actions.

To model the current and proposed SoS SE processes in terms of time taken to complete a SoS program, the probability of success for selected activities needs to be determined and assigned. From the findings in (Reig *et al.*, 1999), 62% of 33 programs with EMD ending between 1980 and 1996 had schedule overruns. Using this information, this work assumes the probability of failure of Sys5 (Verify system's satisfaction of SoS requirements) to be 0.62. The probability of failure for SoS11 (Test and Validation of SoS) can be assumed to be higher than that of Sys5. This is because a SoS is more complex and has many more interfaces and interoperability issues to resolve than does a system. Therefore, taking a conservative approach, the probability of failure for SoS11 is set equal to that of Sys5. In addition, owing to a lack of established data, the probability of success for the activities with feedback in the proposed SoS SE process (Table 3) is set at 0.5 (i.e., there is an equal probability that the activity may pass or fail). Setting the probability of success and failure to be equal is a very conservative way to ensure there is no bias towards the proposed SoS SE process.

Table 3: Probability of Success for Activities with Feedback

S/N	Activity	Probability of Success	
		Current	Proposed
SoS2	Verify and Validate SoS Requirements*	-	0.5
SoS5	Generate SoS Concept Alternatives*	-	0.5
SoS7	Generate SoS Architecture Alternatives*	-	0.5
SoS8	Select Final SoS Architecture*	-	0.5
SoS9	Design SoS*	-	0.5
SoS11	Test and Validate SoS	0.38	0.5
Sys5	Verify System's Satisfaction of SoS Requirements	0.38	0.5

V. DISCUSSION OF RESULTS

This chapter focuses on discussing and analyzing the M&S results. The key measure of effectiveness (MOE) used in the comparison of the current and proposed SoS SE processes is the time taken to complete a SoS acquisition program. A sensitivity analysis is also included to study the effects of varying probability values on the time taken to complete the program. The chapter is divided into three sections. Section A looks into the mean activity processing time obtained using the raw data output from the simulation. Section B compares the time taken to complete the SoS acquisition program for the current and proposed SoS SE processes, while Section C discusses the sensitivity analysis.

A. MEAN ACTIVITY PROCESSING TIME

The mean activity processing times for a three-system SoS based on the current and proposed SoS SE processes are shown in Figures 23 and 24, respectively. The mean activity completion time is obtained by taking the mean of the raw processing times obtained from the simulation. The trends observed in the mean activity processing time will be explained using the structure and dynamics of the ExtendSim model and the SysML activity diagrams.

Figure 23 shows that the activity with the longest mean activity processing time for the current SoS SE process is Sys4 (Implement Changes/Modifications to Systems). There are two possible contributing factors for this observation.

1. Number of Feedback Loops

Sys4 is the only activity that has two feedback loops linked to it. They are from Sys5 (Verify system's satisfaction of SoS requirements) and SoS11 (Test and Validate SoS). This increases the chance of Sys4 being performed a number of times as a result of potential failures in Sys5 and SoS11.

2. Mean Processing Time

Table 2 shows that Sys4 has one of the longest mean processing times (5.7 time units) compared with the other activities in the current SoS SE process. Sys4 represents implementing changes and modifications to systems, which is similar to assembly and production activities. If there is a need to redo Sys4 several times, as brought about by feedbacks from Sys5 and SoS11, the total time taken for completing the program may increase significantly.

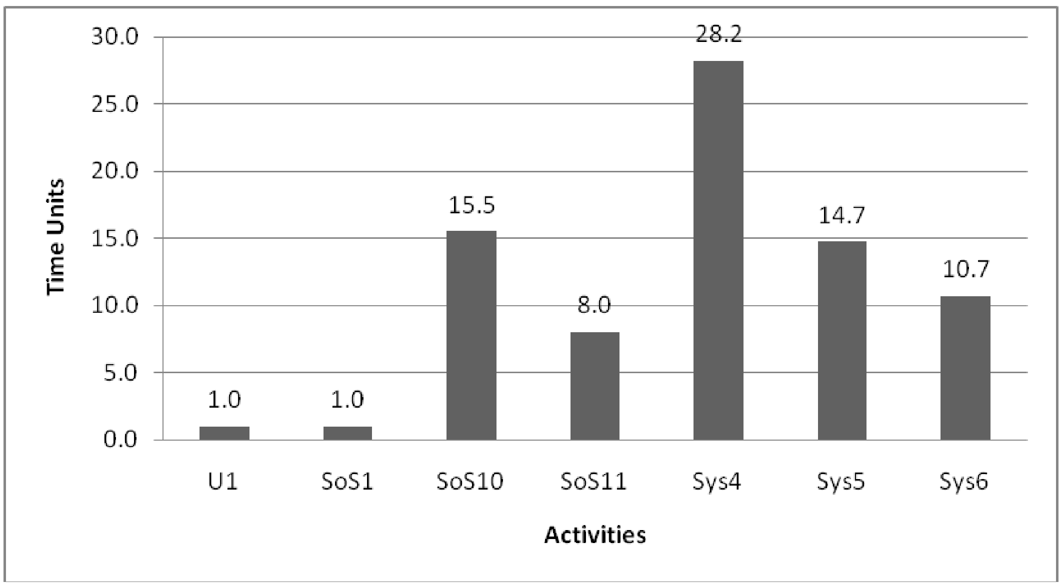

Figure 23: Mean Processing Time for Activities in the Current SoS SE Process

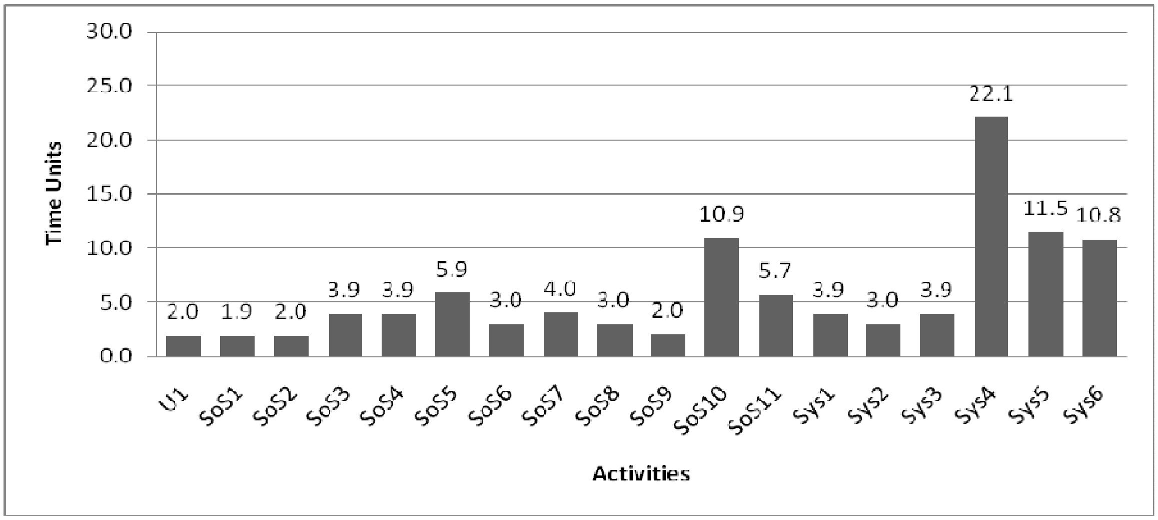

Figure 24: Mean Processing Time for Activities in the Proposed SoS SE Process

Similarly, Figure 24 shows that Sys4 is the activity with the longest mean activity processing time. The ExtendSim model and the SysML activity diagram show that, unlike Sys4 in the proposed SoS SE process which is not linked to feedback loops, Sys3 (Define Changes/Modifications to System Based on SoS Design) is linked to two feedback loops from Sys5 and SoS11. Coupled with a longer mean processing time, and the fact that whatever passes through Sys3 also passes through Sys4, the mean activity processing time for Sys4 can be expected to be the longest of all the activities.

Next, the mean activity processing times for the current and proposed SoS SE processes are compared. Examining the activities that are common to both SoS SE processes reveals that the proposed SoS SE process is able to reduce the individual mean activity processing time for SoS10, SoS11, and Sys4 through Sys6 by about 21.6% to 29.7%. Activities U1 and SoS1 see a 90%-100% increase in their mean activity processing times. However, the absolute time increase is not significant and the reason for the increase could be the feedback introduced into the proposed SoS SE process (SoS2 to U1), which increases the chance of both activities being executed again because of possible failures to verify or validate. The improvement for Sys6 is marginal when comparing the current and proposed SoS SE processes. Table 4 lists the mean activity processing times and the percentage improvement offered by the proposed SoS SE process over the current SoS SE process.

Table 4: Comparison of Common Activities between Current and Proposed SoS SE Processes

Activity	Current (time units)	Proposed (time units)	Percent Improvement (%)
U1	1.0	2.0	-100
SoS1	1.0	1.9	-90
SoS10	15.5	10.9	29.7
SoS11	8.0	5.7	28.8
Sys4	28.2	22.1	21.6
Sys5	14.7	11.5	21.8
Sys6	10.7	10.8	-0.9
Total	**79.1**	**64.9**	

B. CUMULATIVE PROCESSING TIME

The cumulative processing time is the total time required to complete the SoS acquisition. As revealed by Figures 25 and 26, the cumulative processing times for the current and proposed SoS SE processes are 79.1 and 103.4, respectively. In addition, both figures also display the flow sequence of the activities in each of the SE processes being studied.

As discussed in Section A, because of the realization of the proposed front-end SE activities, the mean activity processing time for the key activities (Sys4, Sys5, SoS10 and SoS11) in the current SoS SE process is now reduced by between 21.6% and 29.7%.

Whereas the key activities enjoy such a processing time reduction, the front-end SE activities may increase the overall program completion time. The program completion time can be significantly affected by the probabilities of failure or success of the activities with feedback (Table 3). Since the values of these probabilities are assumed in this simulative study, a sensitivity analysis is conducted to assess the effects of different probability values on the program completion time.

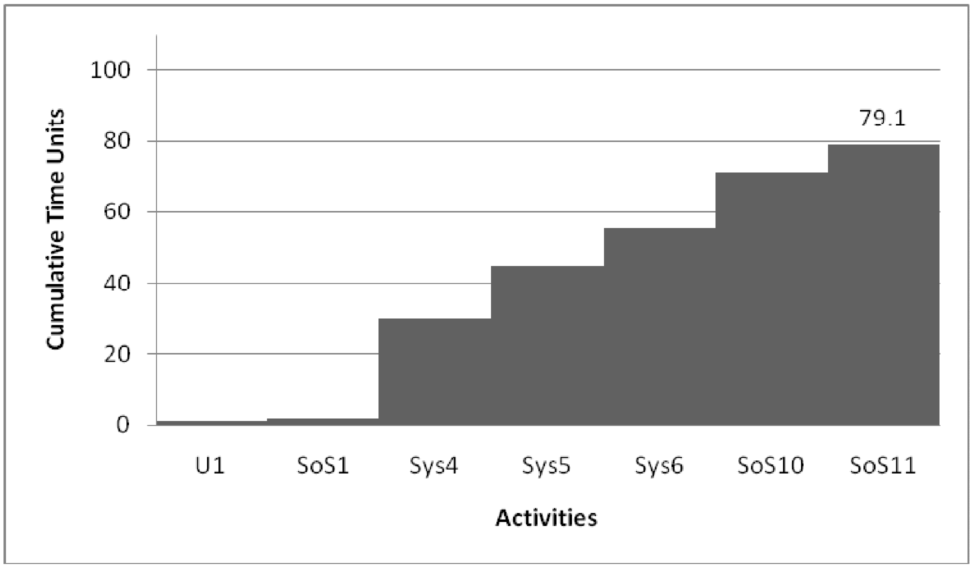

Figure 25: Cumulative Processing Time for Current SoS SE Process

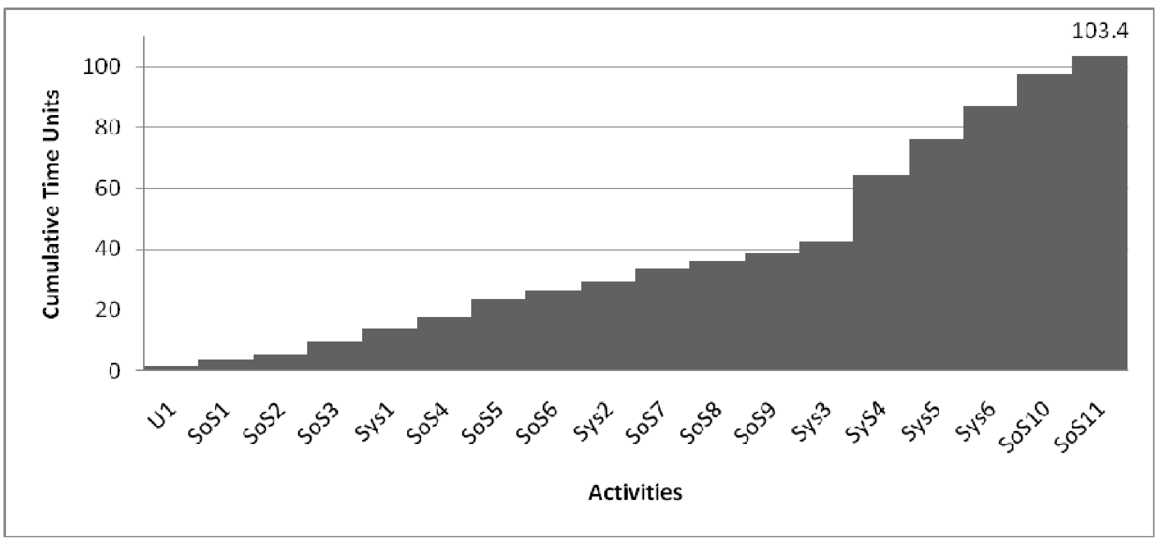

Figure 26: Cumulative Processing Time for Proposed SoS SE Process

C. SENSITIVITY ANALYSIS

Two sensitivity studies are conducted to assess the sensitivity of the total program completion time to (1) the probabilities of failure of the verification, test, and validation activities; and (2) the probabilities of failure of the front-end SE activities.

1. Study 1: Effects of Probabilities of Failure for Verification, Test, and Validation Activities on Program Completion Time

In this study, the probability of failure (Pf) of 0.5 is assumed for all activities with feedback. This is a conservative assumption, because the well-executed front-end SE activities would lower the Pf of the verification, test, and validation activities. The Pf of Sys5 (Verify system's satisfaction of SoS requirements) and SoS11 (Test and Validate SoS) in the proposed SoS SE process is varied from about 0.2 to 0.5. This range of Pf values corresponds to a decrease of 30% to 80% in the baseline-case Pf. The baseline-case Pf is one in which the Pf of Sys5 and SoS11 in the current SoS SE process is 0.62 (Reig *et al.*, 1999). For each value of the Pf of Sys5 and SoS11 − 0.5, 0.62 and 0.8, which represent the different risk levels in the current SoS SE process, the percent improvement in the program completion time brought about by the proposed SoS SE process is obtained.

Figure 27 shows the percent improvement in the program completion time when the proposed SoS SE process is used. If the Pf of Sys5 and SoS11 in the proposed SoS SE process is reduced by 50% (i.e., Pf = 0.31) from the baseline Pf, then there is no change in the total program completion time. This value of Pf may be achievable if the SoS and individual system program offices diligently carry out the front-end SE activities.

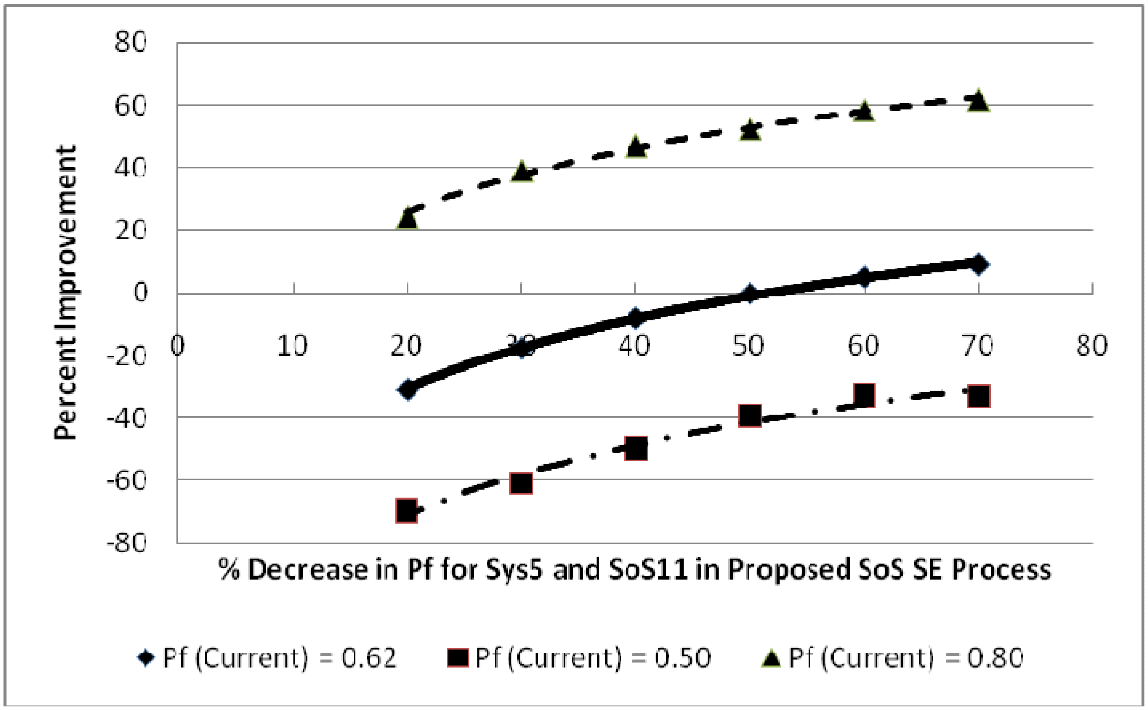

Figure 27: Percent Improvement in Program Completion Time for Different Probability of Failures in Current SoS SE Process (Decrease Probability of Failure for Sys 5 and SoS11 in Proposed SoS SE Process)

As Figure 27 shows, if the Pf is low (e.g., Pf(current) = 0.5), there is no improvement at all (i.e. negative percent improvement). If the Pf is high (e.g., Pf(current) = 0.8), the percent improvement over the current SoS SE process when the proposed SoS SE process is used is about 24% to 61%.

Thus, if the current SoS SE process with a probability of failure of 0.5 (i.e., Pf(current) = 0.5), the SoS program office could continue to use the current SoS SE process. However, if the probability of failure for the SoS acquisition program is high (i.e., Pf(current) = 0.8), the SoS program office should use the proposed SoS SE process as a mitigating measure to reduce the chance of schedule overruns.

2. Study 2: Effects of Probabilities of Failure for Front-End SE Activities on Program Completion Time

This study is focused on the probabilities of failure for the front-end SE activities in the proposed SoS SE process. In the baseline case discussed in Chapter IV, the

probabilities of failure (Pf) for these front-end SE activities are fixed at 0.5 (i.e., equal probabilities of failure and success for these activities). This is a very conservative assumption, because most of these activities would not likely to fail if the front-end activities were efficiently and systematically carried out. In this sensitivity analysis, the probabilities of success (Ps) of the five front-end SE activities with feedback (SoS2, SoS5, SoS7, SoS8, and SoS 9) are incrementally varied from 0.5 to 0.9. Figure 28 shows the variation in the percent improvement in the program completion time offered by the proposed SoS SE process for different Pf (current) values of 0.5, 0.62, and 0.8.

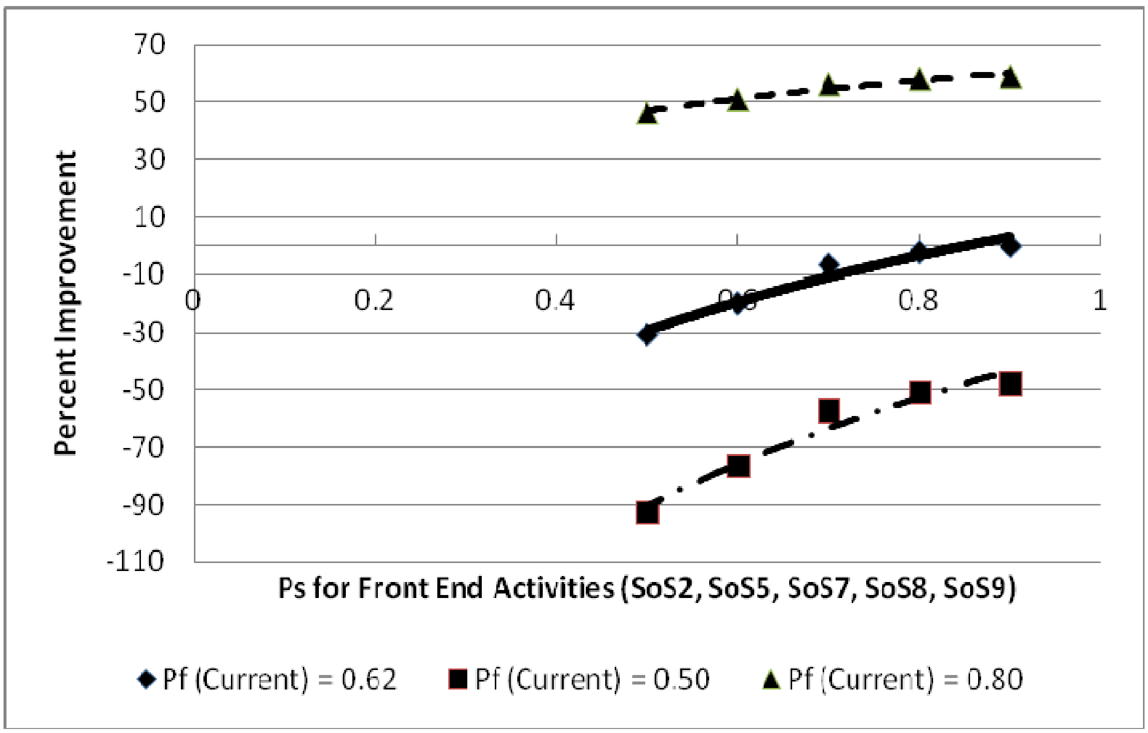

Figure 28: Percent Improvement in Program Completion Time for Different Probability of Failures in Current SoS SE Process (Decrease Probability of Failure for Front-End Activities in Proposed SoS SE Process)

Figure 28 shows that, when Pf (current) is less than 0.62, increasing the Ps for the five front-end SE activities does not lead to an improvement in the program completion time of the acquisition program. However, when Pf (current) is 0.8, the percent improvement in the program completion time is significant, even if the Ps of the five front-end SE activities remain at 0.5.

In wrapping up the sensitivity analysis, when Pf (current) is 0.8, the improvement in the program completion time brought about by the proposed SoS SE process is more sensitive to the Pf of Sys5 and SosS11 than to the Ps of the front-end SE activities.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. SUMMARY / CONCLUSION

This chapter summarizes the key findings from this research and the conclusions drawn from these findings on enhancing the success of SoS acquisition programs.

A. RESEARCH SUMMARY

This research aims to enhance the success of SoS acquisition in DoD by proposing a SoS SE process that includes a set of front-end SE activities to be carried out before the actual development of the SoS commences. M&S is used to provide a quantitative comparison between the current and proposed SoS SE processes. This section provides a summary of the answers to the research questions posed in Chapter I.

1. Current Issues Affecting Success of SoS Acquisition Programs

The current SoS SE process does not have a comprehensive span-of-control over the development of the SoS. This may have led to many of the technical and program management challenges facing the program offices. The proposed SoS SE process would help to provide span-of-control to the SoS program office.

2. Differences Between Proposed SE Process and Current SE Process Used in SoS Acquisition

The current SoS SE process does not have a comprehensive feedback and system architecting mechanism that can help reduce the probabilities of failure of key activities such as verification of systems and test and validation of SoS. To close this gap, the proposed SoS SE process starts with verifying and validating requirements with the user before embarking on a comprehensive set of front-end SE activities to produce the SoS architecture that serves as a reference in the development of the SoS. The proposed SE process also allows increased interaction between the SoS program office with the individual system program offices throughout the course of development of the SoS.

3. Quantitative Comparison Between Current and Proposed SoS SE Processes

M&S is used in the quantitative comparison between the current and proposed SoS SE processes. The main measure of effectiveness is the time taken to complete the SoS acquisition program. A sensitivity analysis is also performed to analyze the effects of the probabilities of failure or success associated with the various activities in the current and proposed SoS SE processes on the degree of improvement in the program completion time brought about by the proposed SoS SE process over the current SoS SE process. The findings are now described.

B. KEY FINDINGS

This section presents the key findings from this research based on the quantitative results obtained from M&S.

1. Mean Activity Processing Time

The M&S results show that the mean activity processing time for the key activities (i.e., Sys4, Sys5, SoS10, and SoS11) decreases with the implementation of the proposed SoS SE process. This decrease in the mean activity processing time is brought about by a lower probability of failure for the verification, testing, and validation activities (Sys5 and SoS11) in the proposed SoS SE process.

2. Program Completion Time

While the mean activity processing time for the key activities (Sys4, Sys5, SoS10 and SoS11) decreases, the front-end SE activities in the proposed SoS SE process may increase the overall program completion time. However, when the Pf for Sys5 and SoS11 of the current SoS SE process is high (i.e. Pf = 0.8), the overall program completion time is significantly improved by the proposed SoS SE process.

3. Sensitivity Analysis

As the probability values used for the activities in the proposed SoS SE process are based on very conservative assumptions, the sensitivity analysis is performed to aid in understanding the behavior of the program completion time when the probabilities of failure for verification, testing, and validation activities in the current SoS SE process are varied. The M&S results demonstrate that, if the verification, testing, and validation activities in the current SoS SE process have a high probability of failure, the percent improvement in the program completion time brought about by the proposed SoS SE process will be significant. If the verification, testing, and validation activities in the current SoS SE process have a low probability of failure (i.e., less than the baseline value of 0.62), the percent improvement in the program completion time for the proposed SoS SE process will be either negative or insignificant.

Hence, the proposed SoS SE process will have a positive impact on the program completion time if the probability of failure of the program is high (i.e., if Pf(current) is 0.8 or higher), resulting from the application of the current SoS SE process. If the time saved from the verification, testing, and validation activities were not sufficient to compensate for the time taken to perform the front-end SE activities (as seen for the cases when Pf(current) = 0.5 and 0.62), the current SoS SE process could still be applied. However, the demise of the recent SoS acquisitions (Huynh *et al.*, 2011) suggests that the current SoS SE process has not successfully supported these SoS acquisitions. As implied by the findings of this research, the proposed SoS SE process appears to be a remedy for these high-risk SoS acquisitions.

C. CONCLUSION

In conclusion, the proposed SoS SE process is able to reduce the mean activity processing time for the key activities such as implementing changes /modifications to systems, system's verification in satisfying SoS requirements, integrating and synthesizing the SoS, and testing and validation of the SoS. This is possible through the setting up of an independent SoS program office that is able to exercise an extensive span-of-control over the SoS development by having in place a comprehensive series of

front-end SE activities. However, to have a significant improvement in program completion time, the probabilities of failure for verification, testing, and validation activities in the current SoS SE process needs to be high (i.e. Pf(current) = 0.8 or higher). This is because the proposed SoS SE process needs to have a significant time savings in the key activities (i.e., Sys4, Sys5, SoS10 and SoS11) in order to compensate for the time needed for the front-end SE activities. Only then can the proposed SoS SE process have a shorter program completion time than does the current SoS SE process. It is high-risk SoS acquisitions such as the U.S. Army's Future Combat System, the U.S. Coast Guard's Deep Water System, the Joint Tactical Radio System (JTRS), and the Homeland Security's SBInet (Huynh *et al.*, 2010) that would likely benefit from the proposed SoS SE process.

VII. RECOMMENDATIONS / FUTURE WORK

A. CHANGES TO SOS ACQUISITION PROGRAMS

The proposed SoS SE process proposed in this study has been shown to reduce the probability of schedule delays for high-risk SoS acquisition programs. While much coordination, administrative efforts, and time will be required to perform the front-end SE process, the benefits reaped, especially in reducing the number of iterations/rework at the later stages of the acquisition process (i.e., verification, testing, and validation activities) are expected to be significant. In particular, emphasis on the front-end SE activities from the acquisition leadership is crucial to ensure that a good systems engineering approach is adopted throughout the SoS acquisition.

B. FUTURE WORK

This research constitutes an initial step in identifying key areas of process improvement to the SE process in a SoS acquisition. Through M&S, it has been shown that introducing front-end SE activities can reduce the time taken for the key activities such as implementing changes/modifications to systems, system's verification in satisfying SoS requirements, integrating and synthesizing the SoS, and testing and validation of the SoS. In terms of the program completion time, the proposed SoS SE process is sensitive to variations in the probability of failure of verification, testing, and validation activities in the current SoS SE process. The following studies/research are therefore recommended:

1. Use Performance and Budget as MOEs

Time or schedule is used as the MOE in this research. Equally important are performance and budget, which may result in the success or failure of SoS programs. It is therefore recommended that performance and cost parameters be included in the model to provide a complete picture of the benefits of the proposed SoS SE process.

2. Data Collection for Mean Processing Times and Probability of Failure for Each Activity

The mean processing time and the probability of failure for each activity are based on assumptions and survey of several single-system programs. To better reflect the actual performance of the proposed SoS SE process, a comprehensive data collection effort should be carried out on on-going SoS acquisition programs. This will provide a more realistic and current overview of the ability of the proposed SoS SE process in improving the completion time of the SoS program.

3. Application of Proposed SoS SE Process to Actual SoS Acquisitions

In parallel, high-risk SoS programs may adopt the proposed SoS SE process by introducing front-end SE activities to help improve the chance of meeting the program's schedule. This can also help in data collection to validate the results obtained in this research.

APPENDIX A

This appendix lists the Level 2 ExtendSim models used in the current SoS SE process for SoS acquisition.

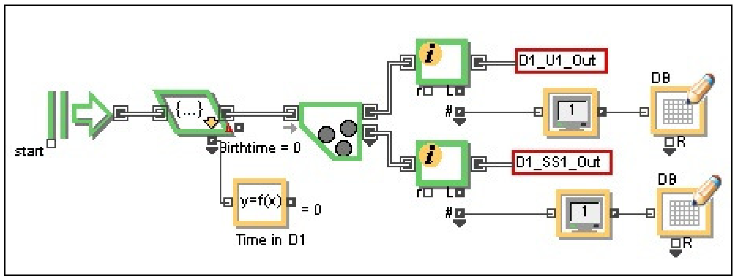

Figure 29: ExtendSim Model for D1: Makes SoS Acquisition Decision (Level 2)

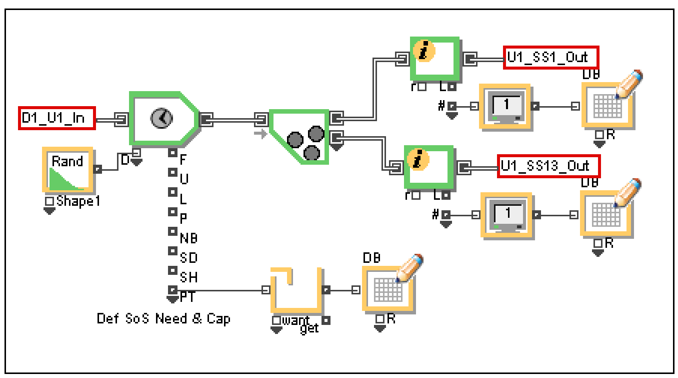

Figure 30: ExtendSim Model for U1: Define Needs and Capabilities (Level 2)

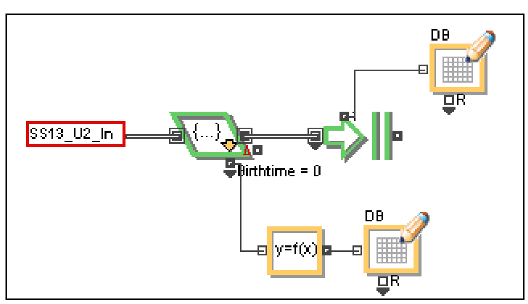

Figure 31: ExtendSim Model for U2: Transit to Operations and Deployment (Level 2)

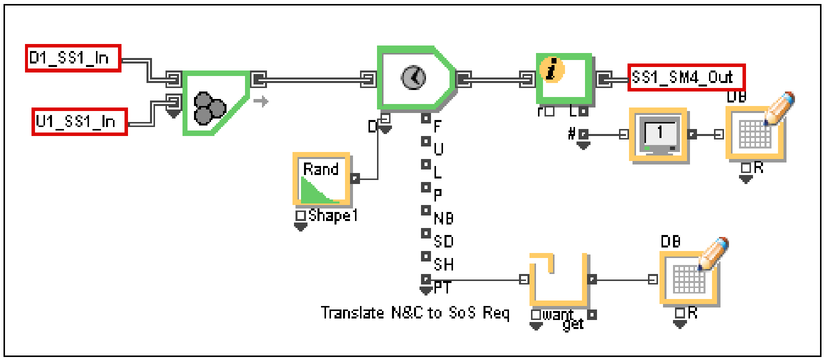

Figure 32: ExtendSim Model for SoS1: Translate Needs and Capabilities to SoS Requirements (Level 2)

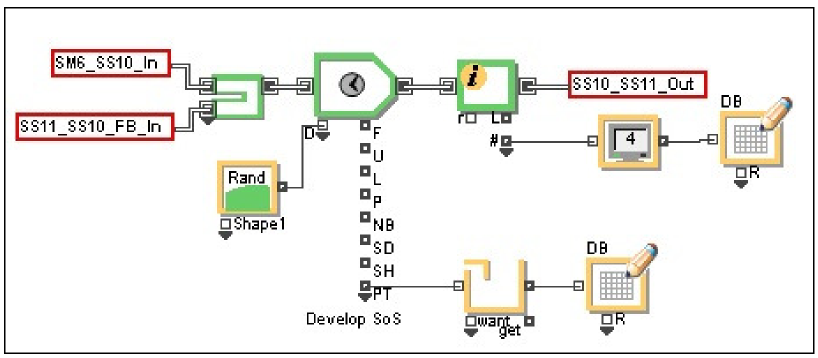

Figure 33: ExtendSim Model for SoS10: Integrate and Synthesize SoS (Level 2)

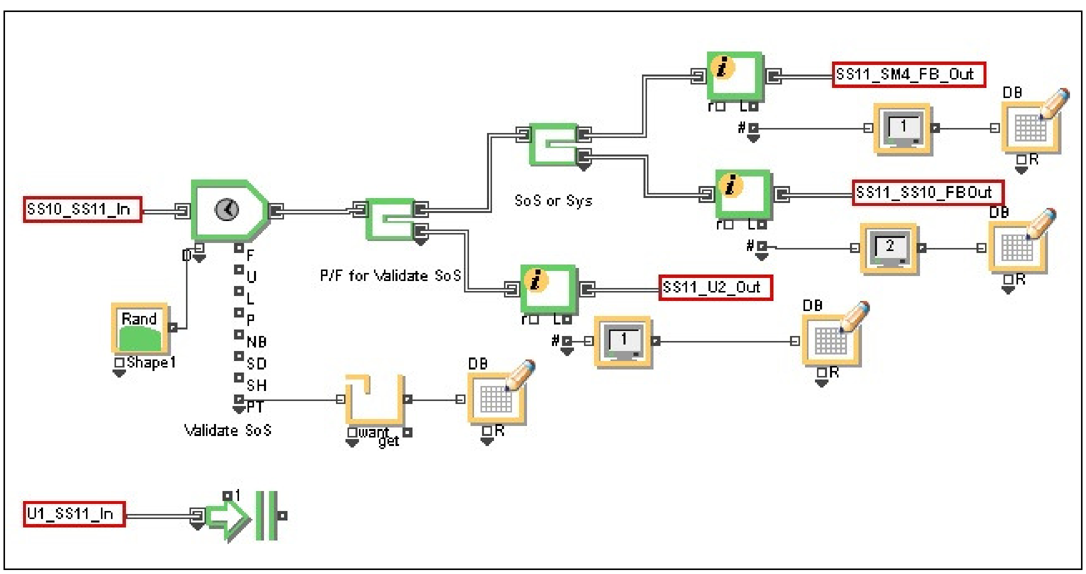

Figure 34: ExtendSim Model for SoS11: Test and Validate SoS (Level 2)

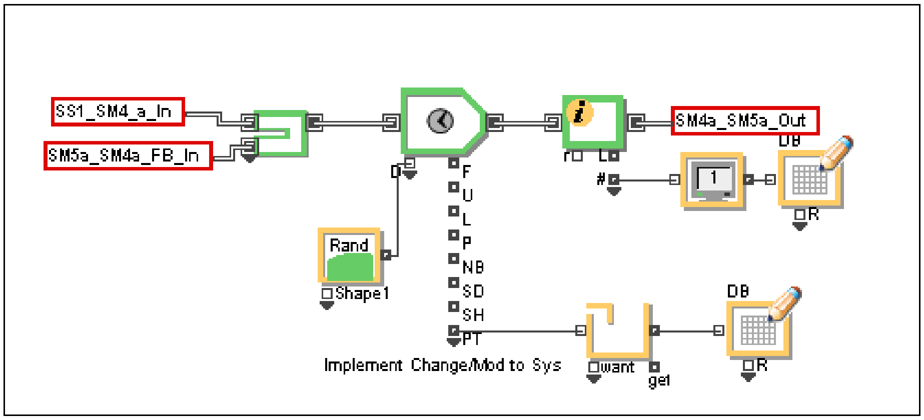

Figure 35: ExtendSim Model for Sys4: Implement Changes/Modifications to System (Level 2)

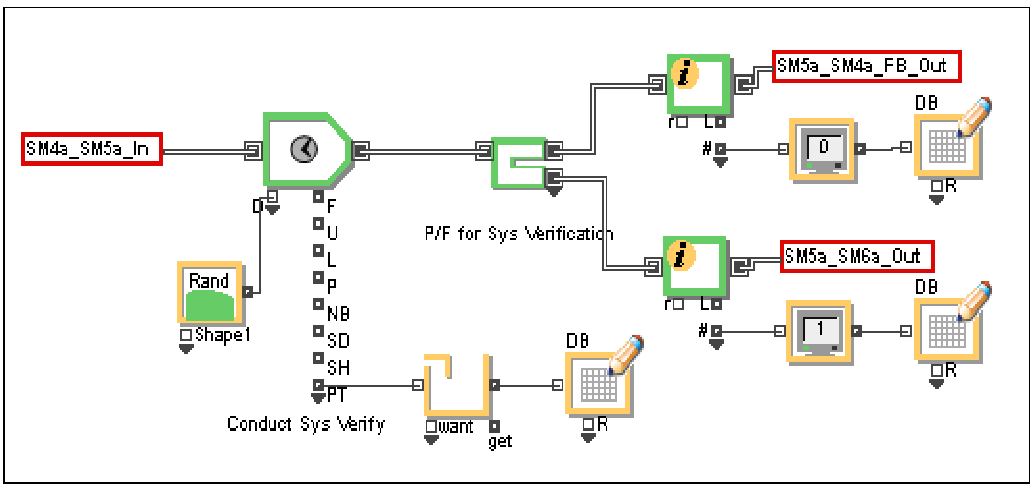

Figure 36: ExtendSim Model for Sys5: Verify System's Satisfaction of SoS Requirments (Level 2)

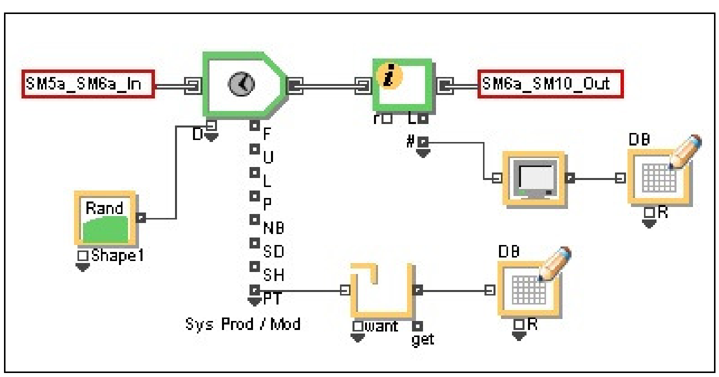

Figure 37: ExtendSim Model for Sys6: System Production / Modification (Level 2)

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B

This appendix lists the Level 2 ExtendSim models used in the proposed SoS SE process for SoS acquisition.

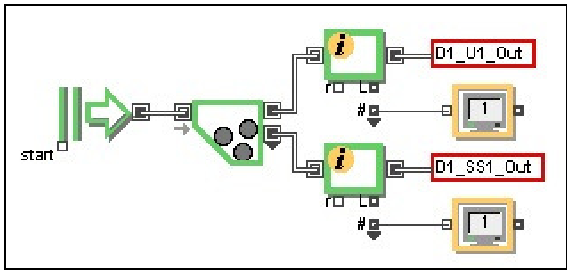

Figure 38: ExtendSim Model for D1: Makes SoS Acquisition Decision (Level 2)

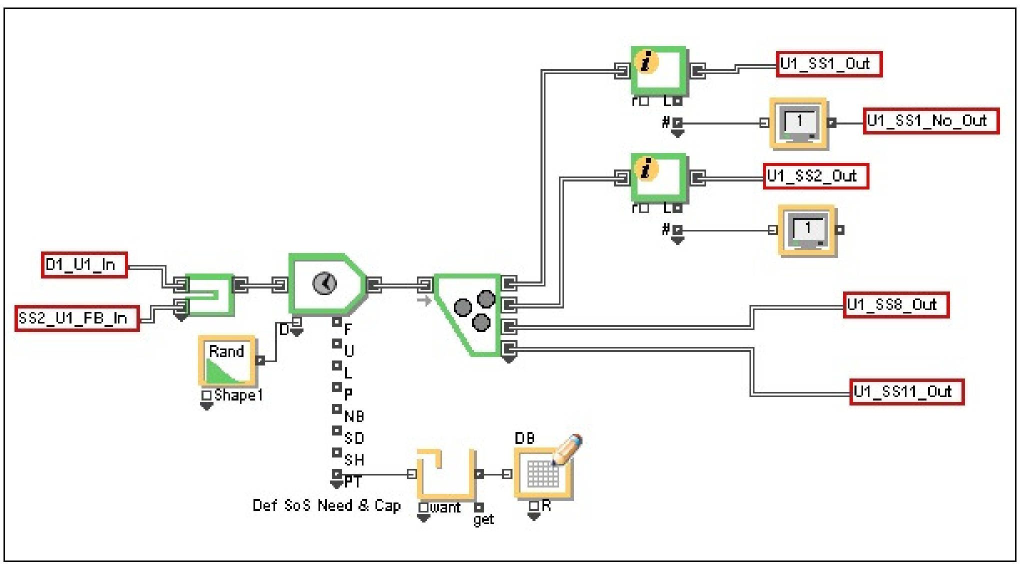

Figure 39: ExtendSim Model for U1: Define SoS Needs and Capabilities (Level 2)

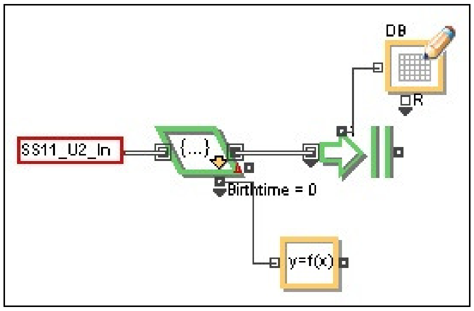

Figure 40: ExtendSim Model for U2: Transit to Operations and Deployment (Level 2)

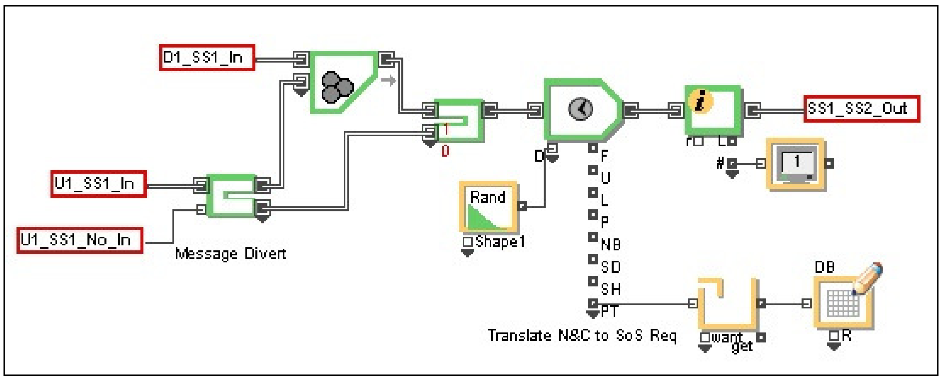

Figure 41: ExtendSim Model for SoS1: Translate Needs and Capabilities into SoS Requirements (Level 2)

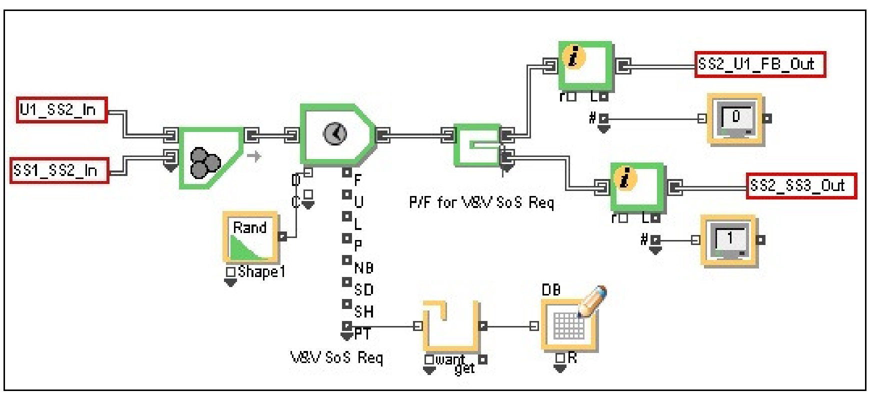

Figure 42: ExtendSim Model for SoS2: Verify and Validate SoS Requirements (Level 2)

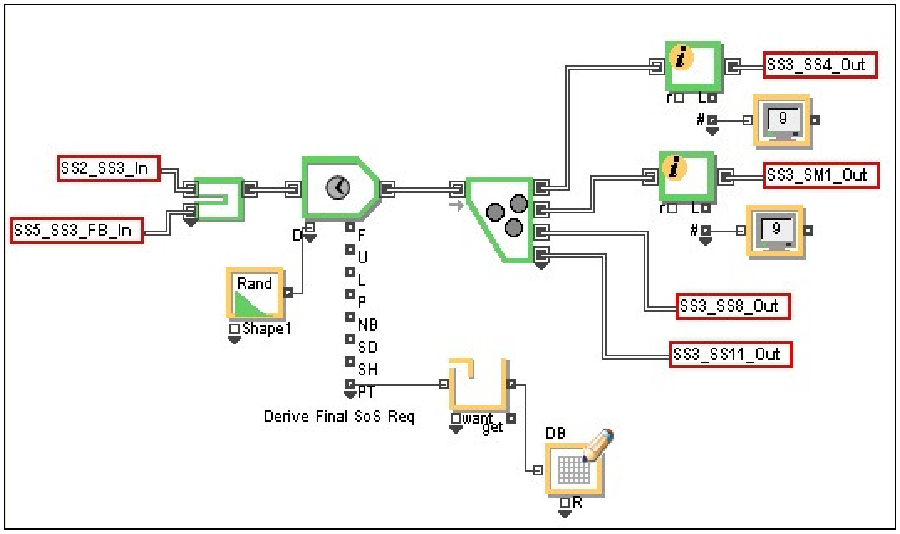

Figure 43: ExtendSim Model for SoS3: Derive Final SoS Requirements (Level 2)

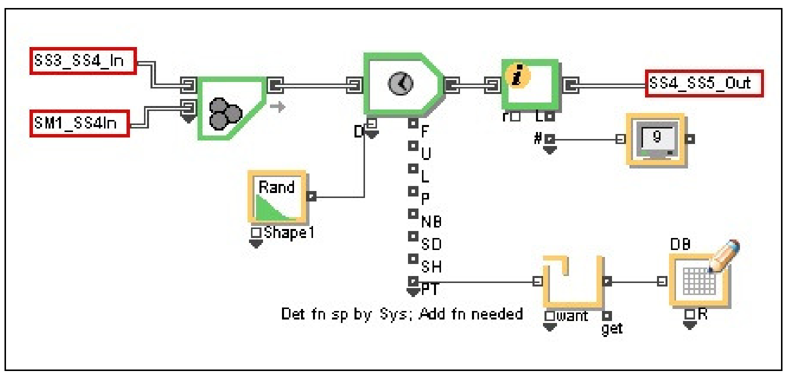

Figure 44: ExtendSim Model for SoS4: Determine SoS Functions Supported by Systems; Additional Functions Needed to Meet SoS Requirements (Level 2)

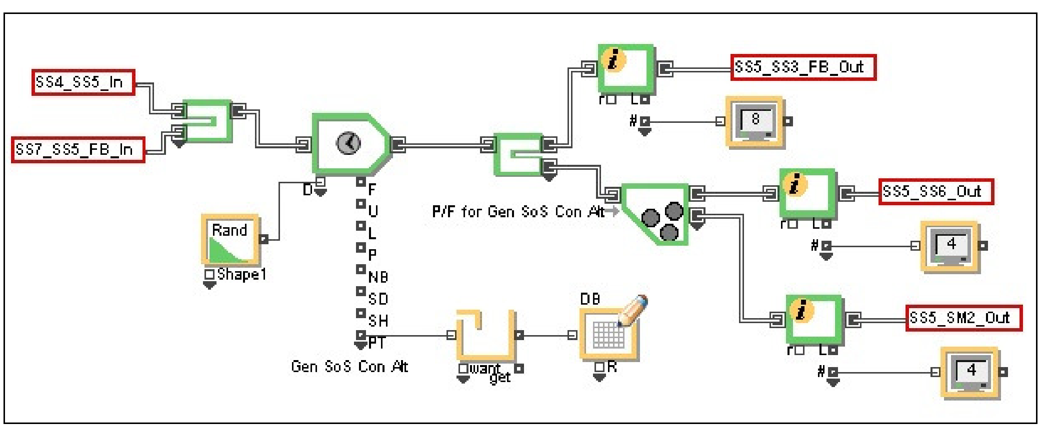

Figure 45: ExtendSim Model for SoS5: Generate SoS Concept Alternatives (Level 2)

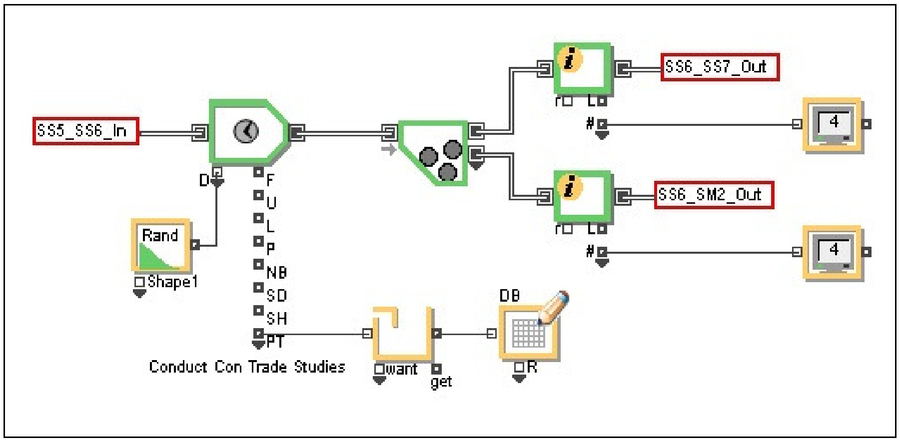

Figure 46: ExtendSim Model for SoS6: Conduct Concepts Trade Studies (Level 2)

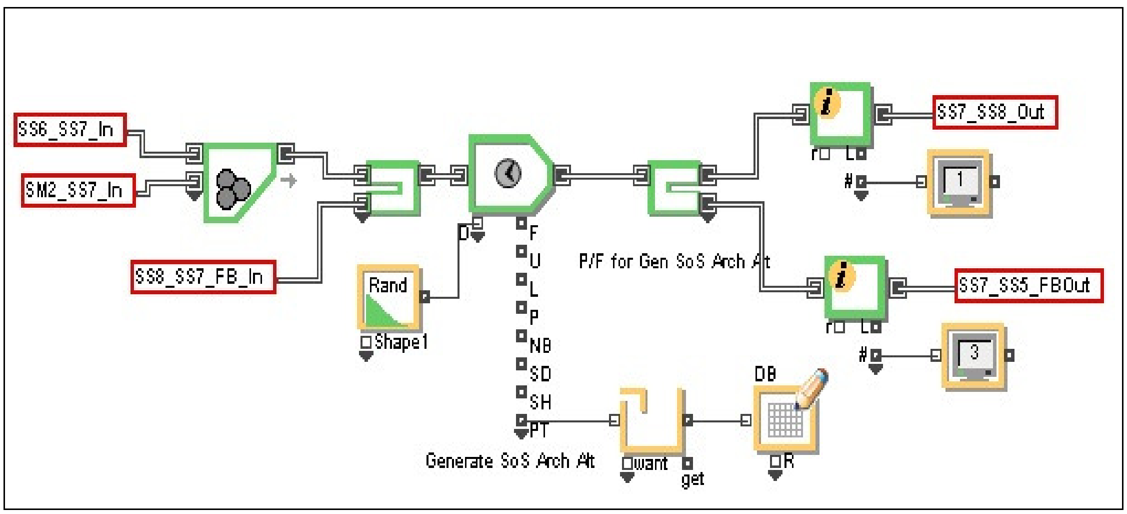

Figure 47: ExtendSim Model for SoS7: Generate SoS Architecture Alternatives (Level 2)

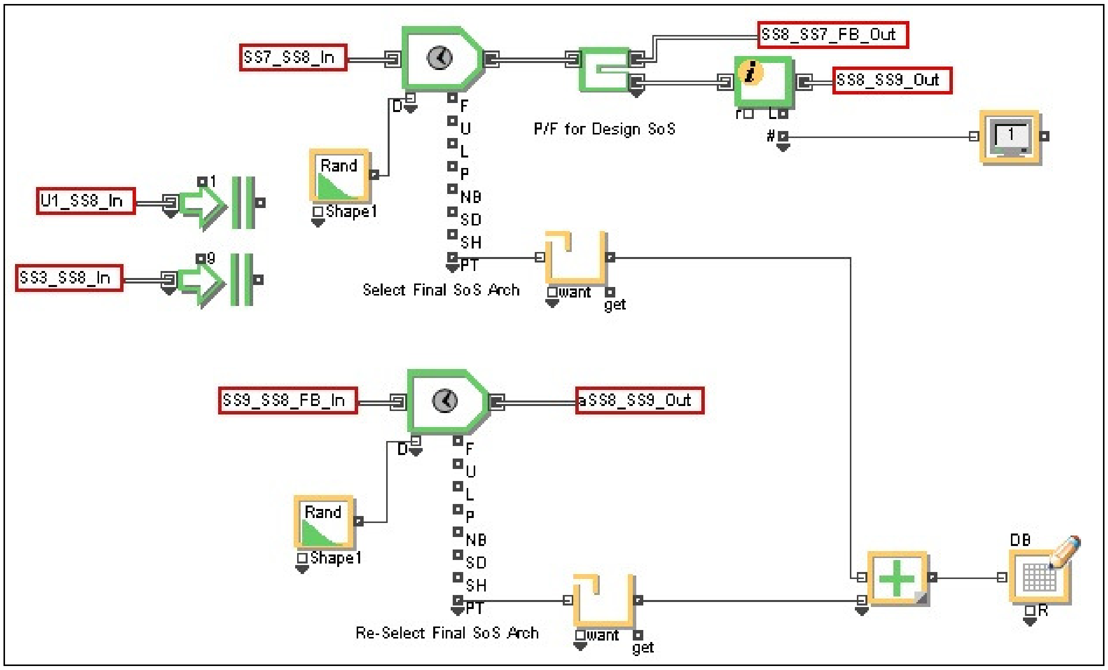

Figure 48: ExtendSim Model for SoS8: Select Final SoS Architecture (Level 2)

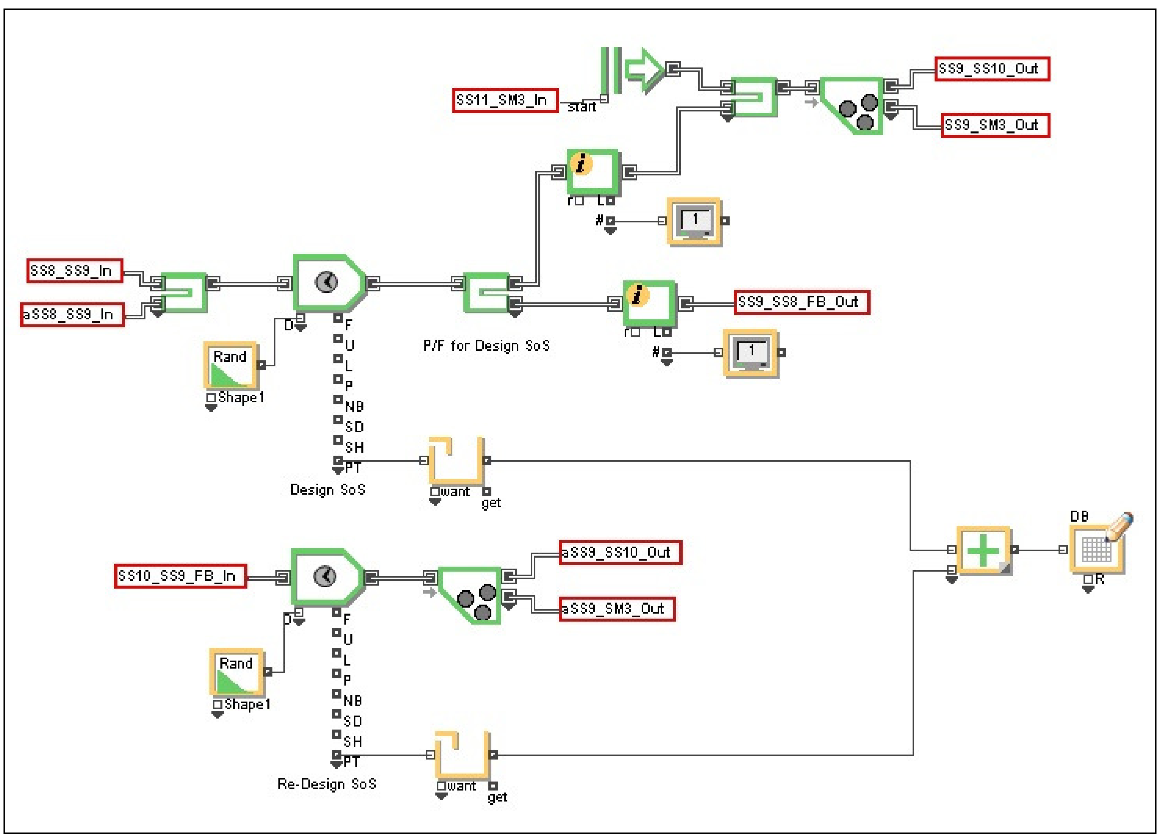

Figure 49: ExtendSim Model for SoS9: Design SoS (Level 2)

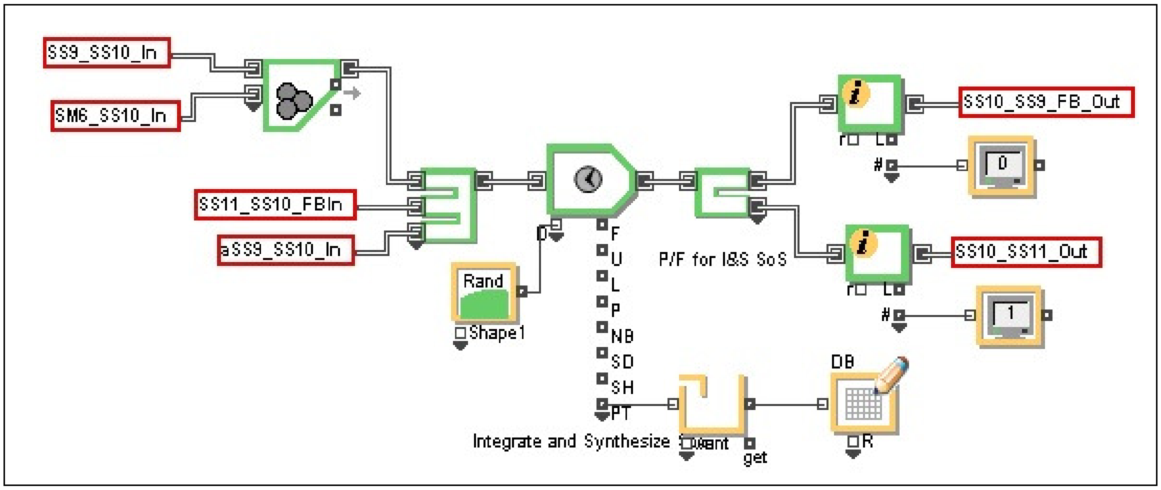

Figure 50: ExtendSim Model for SoS10: Integrate and Synthesize SoS (Level 2)

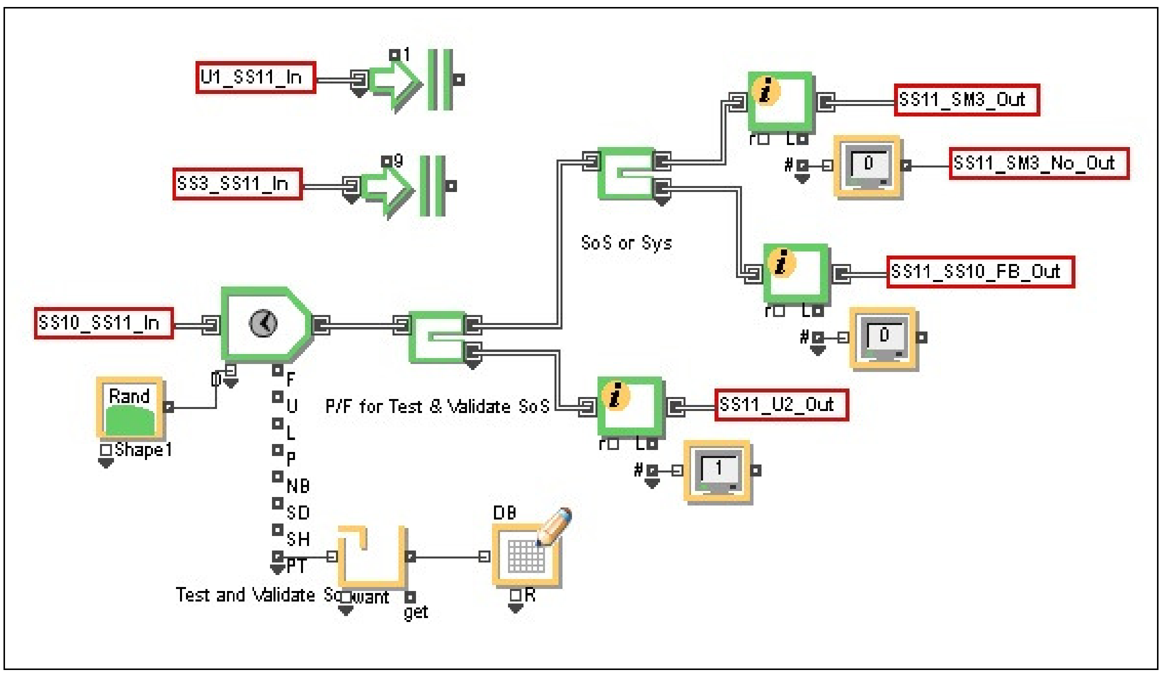

Figure 51: ExtendSim Model for SoS11: Test and Validate SoS (Level 2)

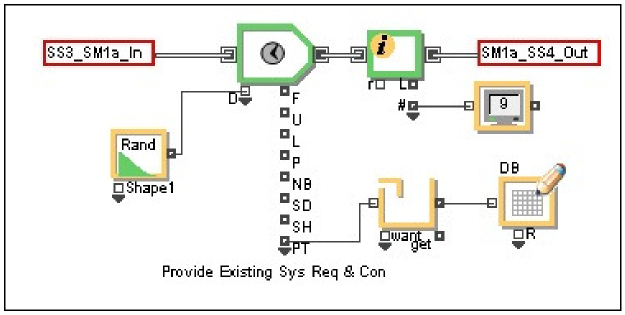

Figure 52: ExtendSim Model for Sys1: Provide Existing Systems Requirements and Constraints (Level 2)

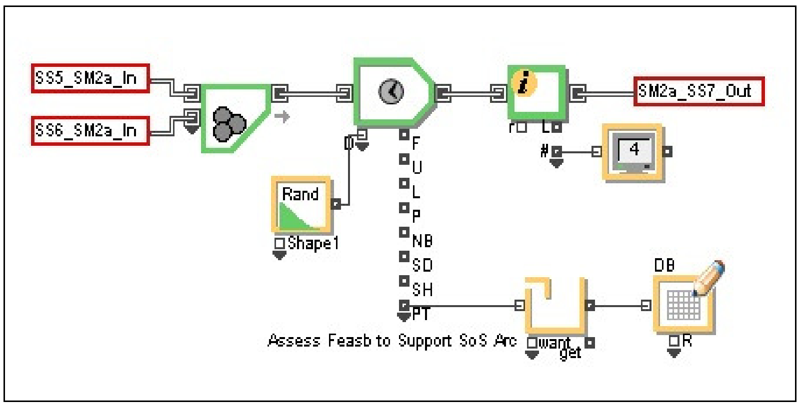

Figure 53: ExtendSim Model for Sys2: Assess Feasibility to Support SoS Architectures (Level 2)

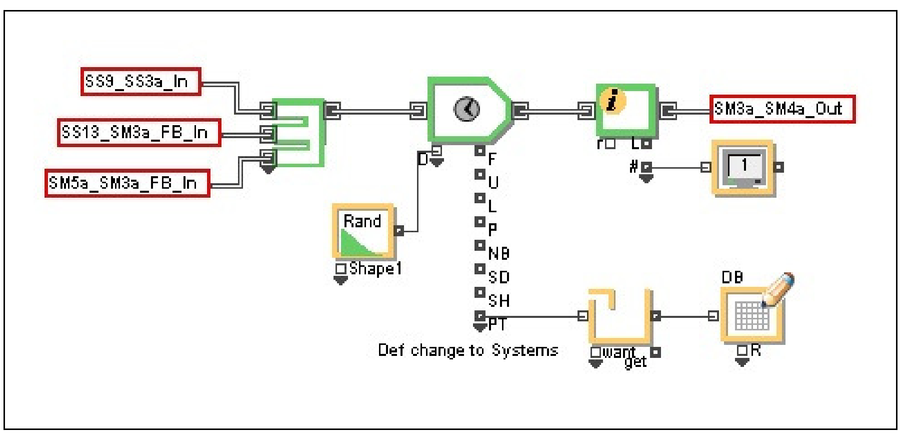

Figure 54: ExtendSim Model for Sys3: Define Changes/Modifications to Systems Based on SoS Design (Level 2)

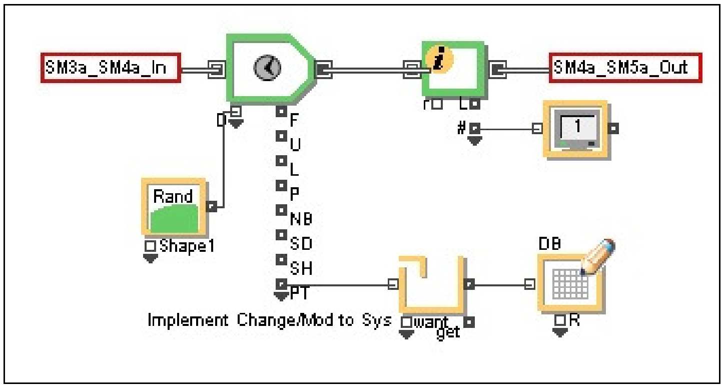

Figure 55: ExtendSim Model for Sys4: Implement Changes / Modifications to Systems (Level 2)

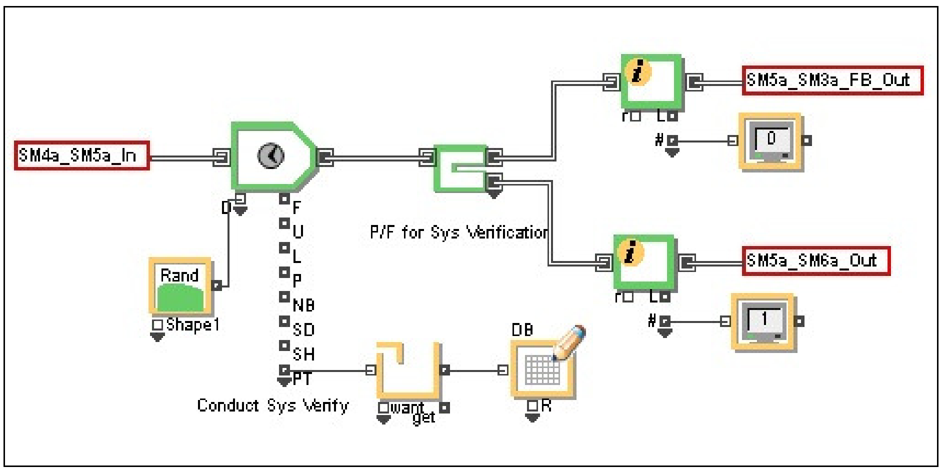

Figure 56: ExtendSim Model for Sys5: Verify System's Satisfaction of SoS Requirements (Level 2)

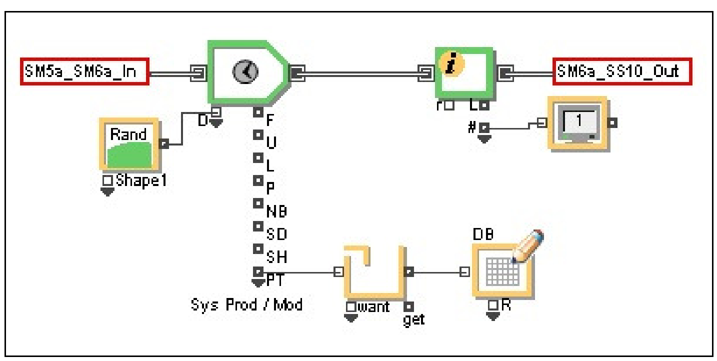

Figure 57: ExtendSim Model for Sys6: System Production / Modification (Level 2)

LIST OF REFERENCES

A. P. Sage & S. M. Biemer. (2007). “Processes for System Family Architecting, Design and Integration,” *IEEE Systems Journal*, *1*(1), 5–16.

D. DeLaurentis & M. Mane. (2009, April 22) “Acquisition Management for Systems-of-systems: Exploratory Model Development and Experimentation,” *Proc. of 6th Annual Acquisition Research Symposium*, pp. 42–59, Naval Postgraduate School, Monterey, CA, Pub. No. NPS-AM-09-013.

Department of Defense (DoD). (2004). Defense Acquisition Guidebook Ch. 4 “System of Systems Engineering," Washington, DC: Pentagon, October 14.

Government Accountability Office (GAO). (2009, March). Defense Acquisition: DoD Must Prioritize its Weapon System Acquisitions and Balance Them with Available Resources (GAO-09-501T). Testimony before the Committee on the Budget, House of Representatives. Retrieved from http://www.gao.gov/new.items/d09501t.pdf

Government Accountability Office (GAO). (2011, March). Defense Acquisitions: Assessments of Selected Weapon Programs (GAO-11-233SP). Report to Congressional Committees. Retrieved from http://www.gao.gov/new.items/d11233sp.pdf

Government Accountability Office (GAO). (1989, August 18). Military Airlift: C-17 Faces Schedule, Cost, and Performance Challenges (NSIAD-89-195). Retrieved from http://archive.gao.gov/d26t7/139358.pdf

Government Accountability Office (GAO). (2000, May). Schedule for Navy Theater Wide Program Should Be Revised to Reduce Risk (GAO/NSIAD-00-121). Retrieved from http://www.gao.gov/archive/2000/ns00121.pdf

Government Accountability Office (GAO). (2011, April). Defense Acquisitions: CH-53K Helicopter Program Has Addressed Early Difficulties and Adopted Strategies to Address Future Risks (GAO-11-332). Retrieved from http://www.gao.gov/new.items/d11332.pdf

Government Accountability Office (GAO). (2005, March). Defense Acquisitions: Changes in E-10A Acquisition Strategy Needed before Development Starts (GAO-05-273). Retrieved from http://www.gao.gov/new.items/d05273.pdf

Government Accountability Office (GAO). (2004, September). Defense Acquisitions: Challenges Facing the DD(X) Destroyer Program (GAO-04-973). Retrieved from http://www.gao.gov/new.items/d04973.pdf

Government Accountability Office (GAO). (2010, October). Defense Acquisitions: DoD Needs to Reassess Joint Cruise Missile Costs before Starting New Production Phase (GAO-11-112). Retrieved from http://www.gao.gov/new.items/d11112.pdf

Government Accountability Office (GAO). (2000, July). Defense Acquisitions: Howitzer Program Experiencing Cost Increases and Schedule Delays (GAO/NSIAD-00-182). Retrieved from http://www.gao.gov/archive/2000/ns00182.pdf

Government Accountability Office (GAO). (2002, July 24). 2002 Update of the 155mm Lightweight Howitzer (GAO-02-898R). Retrieved from http://www.gao.gov/new.items/d02898r.pdf

Government Accountability Office (GAO). (1991, November). Longbow Apache Helicopter: Key Factors Used to Measure Progress in Development Need to be Changed (GAO/NSIAD-92-43) Retrieved from http://archive.gao.gov/d31t10/145486.pdf

Government Accountability Office (GAO). (1988, December). Light Helicopter Program: Risks Facing the Program Raise Doubts About the Army's Acquisition Strategy (GAO/NSIAD-89-72). Retrieved from http://archive.gao.gov/d17t6/137720.pdf

Government Accountability Office (GAO). (2010, July 2). Expeditionary Fighting Vehicle (EFV): Program Faces Cost, Schedule, and Performance Risks (GAO-10-758R). Retrieved from http://www.gao.gov/new.items/d10758r.pdf

Law, A. M. (2007). *Simulation Modeling and Analysis (4th Edition)*, McGraw-Hill, New York, 2007.

Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering. (2008, August). Systems Engineering Guide for Systems of Systems, Version 1.0. Washington D.C: ODUSD(A&T)SSE.

Reig, R. W., Gailey, C. K., Swank, W. J., Alfieri, P.A., & Suycott, M. L.(October 1999). Department of Defense Acquisition Management Metrics, DSMC Press Technical Report TR 1-99, Belvoir, VA.

SysML Partners. (2010, June) OMG SysML Specifications (OMG SysML ™), v. 1.2. Retrieved from: http://www.sysml.org/docs/specs/OMGSysML-v1.2-10-06-02.pdf

T. V. Huynh & J. S. Osmundson. (2007) "An Integrated Systems Engineering Methodology for Analyzing Systems of Systems Architectures," Asia-Pacific Systems Engineering Conference, Singapore, March 23–24, 2007.

T. V. Huynh, R.G. Rendon, & J. S. (2007) Osmundson, “Research on Systems-of-Systems Acquisition,” *Proc. of 7th Annual Acquisition Research Symposium,* pp. 821–842, Naval Postgraduate School, Monterey, CA, Pub. No. NPS-AM-10-071, 2010.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Thomas V. Huynh, PhD
   Department of Systems Engineering
   Naval Postgraduate School
   Monterey, California

4. John S. Osmundson, PhD
   Department of Information Science
   Naval Postgraduate School
   Monterey, California

5. Heng Jiin Shyang
   Defence Science & Technology Agency
   Singapore